HICKMAN PALERMO TRUONG & BECKER LLP
1600 Willow Street
San Jose, CA 95125-5106
(408) 414-1080
Facsimile (408) 414-1076

RECEIVED
CENTRAL FAX CENTER
APR 1 2 2004
OFFICIAL

FACSIMILE

FROM:

Patent Agent:	David Lewis	Direct Phone:	408-414-1080 ext 202
Agent's E-Mail:	dlewis@hptb-law.com	Sender's Fax:	San Jose, CA (408) 414-1076
By Secretary:	Teresa Austin	Direct Phone:	408-414-1080 ext 217
Re:	**Reply to Office Action**	Date: April 8, 2004	Time Sent:

Serial No.: 09/524,725, filed March 14, 2000 Number of pages including this page: 45

TO:

Name	Company	Facsimile No.	Contact No.
Christopher M. Swickhamer	**United States Patent and Trademark Office, GAU 2662**	**703-872-9306**	**703-306-4820**

Pursuant your request we fax the following documents to the above number, we attach copies of the following documents previously faxed to you on April 2, 2004.

ENCLOSURES:

1) Reply to Office Action (7 pgs)
2) Interview Summary (3 pgs)
3) Copy of Return Acknowledgment Postcard (showing receipt by United States Patent and Trademark Office on December 1, 2003) (1 pg)
4) Copy of check in the amount of $144.00
5) Copy of Amendment and Response Transmittal (2 pgs)
6) Copy of Reply to Office Action (15 pgs)
7) Copy of Declaration of Inventor (4 pgs)
8) Copy of Disclosure Document including cover sheet (10 pgs)

CERTIFICATE OF TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted to the U.S. Patent and Trademark Office Fax No. (703) 746-9571.

On April 2, 2004 By Teresa Austin

Teresa Austin

REMARKS

The Examiner is thanked for his conversation with the Applicant's representative on February 26, and 27, 2004 and April 1, 2004, and for indicating that claims 6-9 and 19-22 contain allowable subject matter, removing all of the art rejections of the previous Office Action.

Attached are an Interview Summary, the response filed on September 25, 2003 and faxed again on February 26, 2004, and a coversheet for the facsimile of February 26, 2004.

STATUS OF CLAIMS

Claims 1- 36 are pending in the case. Claims 1-28 are original claims, and claims 29-36 are new claims. No claims are amended.

CLAIM REJECTIONS – 35 U.S.C. § 102 and § 103

The Examiner rejected claims 1-4, 10-17 and 23-28 under 35 U.S.C. §102(e) as allegedly anticipated by Hsu (U.S. Patent No. 6,363,318). The Examiner also rejected claims 5 and 8 under 35 U.S.C. §103 as allegedly unpatentable over Hsu in view of Chang (U.S. Patent Application Publication No. 2003/0123448). These rejections are respectfully traversed.

Declaration under 37 CFR § 1.131

Attached are (1) a Declaration under 37 CFR § 1.131 and (2) a supporting redacted disclosure document. The Office Action (at page 2) stated,

> The evidence submitted is insufficient to establish a conception of the invention prior to the effective date of the Hsu reference.

However, the declaration and response do not mention anything about "conception of the invention". Instead the Applicant's submitted evidence of and argued that the present invention was reduced to practice prior to the date of Hsu.

The Office Action (at page 2) further states,

> The Virtual Dynamic Backbone Protocol (VDBP): Technical Specification by Ryu et al cited to antedate the Hsu reference is not relevant to the claimed invention. The document does not credit the inventor of the instant application as being the author of the VDBP specification. Further the submitted VDBP document describes an invention separate from the invention disclosed in the instant application.

However, the supporting document filed on September 25, 2003 has the title, "Method of determining a datalink path in a managed network". The Applicant's never filed, "The Virtual Dynamic Backbone Protocol (VDBP): Technical Specification by Ryu et al" cited by the Office Action. Apparently, there was a mix up in the mailroom of the US Patent and Trademark Office, and somehow "The Virtual Dynamic Backbone Protocol (VDBP): Technical Specification by Ryu et al" was placed in the present application instead of the "Method of determining a datalink path in a managed network". Possibly, the mailroom also placed the wrong Declaration or other incorrect documents in the application with the response filed September 25, 2003.

Additionally, as stated in the prior response filed on September 25, 2003, the filing date of Hsu, which is August 31, 1999, is less than one year prior to the filing of the

present Application, which is March 14, 2000, and therefore does not constitute a statutory bar. The effective date of Hsu as a reference is August 31, 1999. The Declaration avers the existence, prior to the effective date of Hsu as a reference, of a working version of subject matter that is an embodiment of the claims. The existence of a working version of the claimed invention constitutes a reduction to practice. Thus, the declaration in-and-of-itself evidences that the claimed invention was conceived and reduced to practice prior to the effective date of Hsu as a reference.

The disclosure document describes subject matter that is an embodiment of the claimed invention, and in page 8, the disclosure document states

> Cisco Use: This method is currently being used in Cisco PathTool to determine the layer 2 path.

Since the Declaration avers that the disclosure document was written before the filing date of Hsu, the disclosure document (which is redacted in accordance with MPEP 715.07, p. 700-231) is further evidence that the claimed invention was reduced to practice prior to the filing date of Hsu. Further, the Declaration avers that the date of the disclosure document is prior to the filing date of Hsu. Thus, the Declaration, when combined with the disclosure document, is a showing of facts that are of a character and weight as to establish reduction to practice prior to the effective filing date of the reference. Accordingly, Hsu should be removed as a reference, and the rejection under 35 USC § 102(e) and § 103 should be withdrawn.

OBJECTION TO CLAIMS 6-9 AND 19-22

Since the rejection of the base claims should be removed, the objection to claims 6-9 and 19-22 as depending upon rejected base claims should also be withdrawn.

NEW CLAIMS

As stated in the prior response filed on September 25, 2003, each of new claims 29-36 depend on one of claims 1 and 14. Since Hsu should be removed as a reference regarding the base claims of 1 and 14, therefore claims 29-36 are also allowable. Additionally, the passages of Hsu cited by the Examiner never disclose or suggest (1) using Hsu's method for monitoring, as recited in claims 29 and 33, (2) using Hsu's method for obtaining diagnostic information, as recited in claims 30 and 34, (3) using Hsu's method for tracing a path at level 2, as recited in claims 31 and 35, or (4) that Hsu's method determines a verifiable path that is in a bridge forwarding table, as recited in claims 32 and 36. Further, the title of Hsu is "Constraint-Based Route ***Selection*** Using Biased Cost" (emphasis added), indicating that Hsu uses his method for selecting a route (that presumably was not previous determined), and not for monitoring, obtaining diagnostic information, tracing a path, or determining a path that could have been verified by a comparison with a bridge forwarding table.

REQUEST FOR A CORRECTED OFFICE ACTION RESTARTING THE PERIOD FOR REPLY

MPEP 710.06 states

> Where … an Office action contains some … defect and this error is called to the attention of the Office within 1 month of the mail date of the action, the Office will restart the previously set period for reply to run from the date the error is corrected, if requested to do so by applicant.

The Office Action contains a defect in that it responds to material not filed by the Applicant. As pointed out in the Interview Summary, in a telephone call on February 26, 2004, which is less than a month after the mailing of the Office Action, the Examiner was

in possession of the wrong document. The correct documents were faxed on February 26, 2004. Along with the correct documents, a copy of the filing receipt, return postcard, and transmittal coversheet were also faxed. The copy of the filing receipt, return postcard, and transmittal coversheet (in conjunction with the page counts on the return postcard) evidence that the correct documents were in fact included in the original response filed on September 25, 2003. Therefore, the Patent and Trademark Office was notified that there was a defect in the Office Action within a month of the mailing of the Office Action, and a corrected Office Action resetting the period for reply should be issued in accordance with MPEP 706.10.

CONCLUSIONS AND MISCELLANEOUS

The Applicant's believe that all issues raised in the Office Action have been addressed and that allowance of the pending claims is appropriate. Entry of the amendments herein and further examination on the merits are respectfully requested.

The Examiner is invited to telephone the undersigned at (408) 414-1213 to discuss any issue that may advance prosecution or any other issues related to this application.

No fee is believed to be due specifically in connection with this Reply. To the extent necessary, Applicant's petition for an extension of time under 37 C.F.R. § 1.136. The Commissioner is authorized to charge any fee that may be due in connection with this Reply to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: April 2, 2004

David Lewis

David Lewis
Reg. No. 33,101

1600 Willow Street
San Jose, California 95125-5106
Telephone No.: (408) 414-1080
Facsimile No.: (408) 414-1076

II 13

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Mehryar Garakani

Serial No.: 09/524,725

Filed: March 14, 2000

For: METHOD OF DETERMINING A DATA LINK PATH IN A MANAGED NETWORK

Group Art Unit No.: 2697

Examiner: Christopher M. Swickhamer

Mail Stop AF
Commissioner of Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Interview Summary

On February 26, and 27, 2004 and April 1, 2004 the Applicant's representative, David Lewis, conferred with Examiner Swickhamer regarding the Declaration, supporting documentation, the Office Action, and whether another communication from the Examiner was forthcoming.

Specifically, on February 26, 2004, the Applicant's representative called the Examiner to discuss the Affidavit and supporting documentation, because the Applicant's representative did not understand the remarks on page 2 of the Office Action. However,

CERTIFICATE OF TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted to the U.S. Patent and Trademark Office Fax No. (703) 746-9571.

On April 2, 2004 By Teresa Austin
Teresa Austin

after a few minutes of conversation it became apparent that the Examiner was in possession of the wrong documents, which was verified by the Examiner reading the title of the supporting document before him, which was "The Virtual Dynamic Backbone Protocol (VDBP): Technical Specification by Ryu et al". That evening a second copy of the documents filed on September 25, 2003 was faxed to the Examiner. Included in the facsimile were a copy of the filing receipt, return postcard (which includes a page count for the each of the documents), and transmittal sheet, which (in conjunction with the page counts) evidence that the documents faxed were filed on September 25, 2003 and *not* "The Virtual Dynamic Backbone Protocol (VDBP): Technical Specification by Ryu et al". The next day the Applicant's representative called the Examiner to verify that the facsimile was received. Before getting off the phone, the Applicant's representative remarked that he would be looking forward to another communication from the Examiner regarding the correct papers filed on September 25, 2003 to which the Applicant's representative thought the Examiner responded in the affirmative.

However, after not receiving any communication from the Examiner, on April 1, 2004, the Applicant's representative called the Examiner to ask if any communication had been sent, or if no communication had been sent, when could one be expected. (Apparently, the Examiner did not recall saying, or did not intend to say, something to the effect that a communication would be forthcoming during the conversation of February 27, 2004.) After consulting with his supervisor the Examiner stated that the facsimile was considered only a draft of a response, and that a formal response was still necessary.

No fee is believed to be due specifically in connection with this Reply. To the extent necessary, Applicant's petition for an extension of time under 37 C.F.R. § 1.136. The Commissioner is authorized to charge any fee that may be due in connection with this Reply to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: April 2, 2004

David Lewis
Reg. No. 33,101

1600 Willow Street
San Jose, California 95125-5106
Telephone No.: (408) 414-1080
Facsimile No.: (408) 414-1076

HICKMAN PALERMO TRUONG & BECKER LLP
1600 Willow Street
San Jose, CA 95125-5106
(408) 414-1080
Facsimile (408) 414-1076

COPY

FAXED 2/26/04 @ 1:57 pm.

FACSIMILE

FROM:

Patent Agent:	David Lewis	Direct Phone:	408-414-1080 ext 202
Agent's E-Mail:	dlewis@hptb-law.com	Sender's Fax:	San Jose, CA (408) 414-1076
By Secretary:	Teresa Austin	Direct Phone:	408-414-1080 ext 217
Re:	**Copy of Reply to Office Action mailed 9/25/03**	Date: February 26, 2004	Time Sent:

Serial No.: 09/524,725, filed March 14, 2000 Number of pages including this page: 34

TO:

Name	Company	Facsimile No.	Contact No.
Christopher M. Swickhamer	**United States Patent and Trademark Office, GAU 2662**	**703-746-9571**	**703-306-4820**

MESSAGE:

Pursuant to our conversation on February 26, 2004, I am attaching another copy of Applicant's Reply to Office Action mailed on November 26, 2003. This reply is responsive to the Office Action mailed September 25, 2003.

ENCLOSURES:

1) Return Acknowledgment Postcard (showing receipt by United States Patent and Trademark Office on December 1, 2003) (1 pg)
2) Copy of check in the amount of $144.00
3) Amendment and Response Transmittal (2 pgs)
4) Reply to Office Action (15 pgs)
5) Declaration of Inventor (4 pgs)
6) Disclosure Document including cover sheet (10 pgs)

COPY

COPY

Acknowledgment Receipt

Acknowledgment Receipt

Mailing Date: November [illegible], 2003 Attorney: DL/aSS

Attorney Docket No.: 50325-0088 First Class Mail

Serial No.: [illegible]

Inventor(s): [illegible]

Title: METHOD OF DETERMINING A DATA LINK PATH IN A MANAGED NETWORK

Documents Enclosed:

1) Reply to Office Action (15 pgs)
2) Declaration of Inventor (4 pgs)
3) Disclosure Document including cover sheet (10 pgs)
4) Amendment and Response Transmittal (2 pgs) (in duplicate)
5) Check in the amount of $144.00
6) Return Acknowledgment Postcard

OIPE

Kindly stamp the receipt date and return to addressee to acknowledge receipt of the above documents.

Processed by docketing
no dates docketed

HERITAGE BANK MMERCE
SAN JOSE, ...113

4113

HICKMAN PALERMO TRUONG & BECKER LLP
PTO PAYMENT ACCOUNT
1600 WILLOW STREET
SAN JOSE, CA 95125

90-4228/1211

11/26/03

PAY TO THE ORDER OF Commissioner for Patents and Trademarks $ 144.00

One hundred forty four DOLLARS

Security features. Details on back.

AUTHORIZED SIGNATURE

MEMO Docket No. 50325-0088

⑈004113⑈ ⑆121142287⑆ 001566660⑈

HICKMAN PALERMO TRUONG & BECKER LLP
PTO PAYMENT ACCOUNT

4113

Commissioner for Patents and Trademarks

Date 11/26/03

Amount $ 144

Docket No. 50325-0088

COPY

HICKMAN PALERMO TRUONG & BECKER LLP
PTO PAYMENT ACCOUNT

4113

Commissioner for Patents and Trademarks

Date

Amount $

Docket No.

COPY

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Mehryar Garakani

Serial No.: 09/524,725

Filed: March 14, 2000

For: METHOD OF DETERMINING A DATA LINK PATH IN A MANAGED NETWORK

Group Art Unit No.: 2697

Examiner: Christopher M. Swickhamer

COMMISSIONER FOR PATENTS
P. O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Transmitted herewith is a Reply to Office Action in the above identified application.

☒ Also attached: Declaration of Inventor and Disclosure Document

☒ Return Receipt Postcard.

The fee has been calculated as shown below:

	NO. OF CLAIMS	HIGHEST PREVIOUSLY PAID FOR	EXTRA CLAIMS	RATE	FEE
Total Claims	36	28	8	$18.00 =	$144.00
Independent Claims	4	4	0	$86.00 =	$0.00
		Multiple claims newly presented			$0.00
		Fee for extension of time			$144.00
		TOTAL FEE DUE			$144.00

☒ Enclosed is a check in the amount of $144.00. Please charge any and all deficiencies to Deposit Account No. 50-1302. An additional copy of this transmittal sheet is submitted herewith.

☒ The Commissioner is hereby authorized to charge payment of any fees associated with

this communication or credit any overpayment, to Deposit Account No. 50-1302, including any filing fees under 37 CFR 1.16 for presentation of extra claims and any patent application processing fees under 37 CFR 1.17.

Respectfully submitted,
HICKMAN PALERMO TRUONG & BECKER LLP

David Lewis
Agent of Record
Registration No. 33,101

1600 Willow Street
San Jose, CA 95125
(408) 414-1080 EAB:ss
Date: November 26, 2003
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, Washington, DC 20231

on November 26, 2003 by Sheila Severinghaus

COPY

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:

Mehryar Garakani

Serial No.: 09/524,725

Filed: March 14, 2000

For: METHOD OF DETERMINING A DATA LINK PATH IN A MANAGED NETWORK

Group Art Unit No.: 2697

Examiner: Christopher M. Swickhamer

Commissioner of Patents
P. O. Box 1450
Alexandria, VA 22313-1450

REPLY TO OFFICE ACTION

Sir:

This is in response to the Office Action mailed September 25, 2003, the shortened statutory period for which runs until December 25, 2003.

Additional Claims and Remarks are presented on separate sheets below.

AMENDMENTS TO THE CLAIMS

1. (Original) A method for determining a logical path in a managed network between a source device and a destination device at a data link layer, the method comprising the computer-implemented steps of:

creating and storing a Connected Group Space representation of network devices based on a topology space representation of the network devices;

identifying an optimized path in the Connected Group Space representation;

transforming the optimized path into the topology space representation; and

creating and storing the optimized path that was transformed into the topology space representation as the data link layer path.

2. (Original) The method as recited in Claim 1, wherein the managed network is a managed IP network.

3. (Original) The method as recited in Claim 1, wherein the step of creating and storing a Connected Group Space representation further comprises the steps of:

identifying a set of Connected Group nodes associated with the Connected Group Space representation;

identifying Connected Group links that connect the Connected Group nodes; and

creating and storing information that represents the Connected Group links.

4. (Original) The method as recited in Claim 1, wherein the step of creating and storing a Connected Group Space representation further comprises the steps of:

identifying a subnet associated with the source device and the destination device;

determining a set of network links that link one or more network devices in the managed network; and

determining an assignment of ports of network devices.

5. (Original) The method as recited in Claim 1, wherein the step of creating and storing a Connected Group Space representation further comprises the steps of:

identifying all Virtual Local Area Networks (VLANs) associated with a subnet associated with the source device and the destination device; and

identifying all Emulated Local Area Networks (ELANs) associated with the subnet.

6. (Original) The method as recited in Claim 1, wherein the step of creating and storing a Connected Group Space representation further comprises the steps of:

creating one Connected Group node for any pairs of interfaces across a point-to-point link in the topology space representation;

creating one Connected Group node for any interfaces of the managed network that are directly connected by virtue of being on a same physical medium;

creating one Connected Group node for LAN Emulation interfaces on a same Emulated Local Area Network (ELAN);

creating one Connected Group node for each internal interface of any network device when the network device has an internal interface;

creating one Connected Group node for the source device;

creating one Connected Group node for the destination device; and

creating one Connected Group node for each user interface on any network device when the network device has a user interface.

7. (Original) The method as recited in Claim 6, further comprising the step of determining Connected Group links between Connected Group nodes in a subnet associated with the source device and the destination device.

8. (Original) The method as recited in Claim 7, further comprising the step of creating one Connected Group link for each pair of interfaces within each network device, wherein each interface is associated with the subnet of the source device and the destination device and is in a forwarding state.

9. (Original) The method as recited in Claim 8, further comprising the step of checking a spanning tree status for each interface within each network device to determine whether the interface is in the forwarding state.

10. (Original) The method as recited in Claim 1, wherein the step of identifying an optimized path in the Connected Group Space representation further comprises the step of finding a shortest path between a Connected Group source node and a Connected Group destination node.

11. (Original) The method as recited in Claim 10, further comprising the step of using a Dijkstra algorithm to find the shortest path between the Connected Group source node and the Connected Group destination node.

12. (Original) The method as recited in Claim 1, wherein the step of transforming the optimized path into the topology space representation further comprises the steps of:

identifying an ordered set of Connected Group nodes associated with the optimized path; and

identifying an ordered set of Connected Group links associated with the ordered set of Connected Group nodes.

13. (Original) The method as recited in Claim 12, further comprising the steps of:

identifying a pair of interfaces associated with each Connected Group link in the ordered set of Connected Group nodes associated with the optimized path; and

generating an ordered set of topology space links from the pairs of interfaces associated with Connected Group links.

14. (Original) A computer-readable medium carrying one or more sequences of instructions for determining a logical path in a managed network between a source device and a destination device at a data link layer, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

creating and storing a Connected Group Space representation of network devices based on a topology space representation of the network devices;

identifying an optimized path in the Connected Group Space representation;

transforming the optimized path into the topology space representation; and

creating and storing the optimized path that was transformed into the topology space representation as the data link layer path.

15. (Original) The computer-readable medium as recited in Claim 14, wherein the managed network is a managed IP network.

16. (Original) The computer-readable medium as recited in Claim 14, wherein the step of creating and storing a Connected Group Space representation further comprises the steps of:

identifying a set of Connected Group nodes associated with the Connected Group Space representation;

identifying Connected Group links that connect the Connected Group nodes; and

creating and storing information that represents the Connected Group links.

17. (Original) The computer-readable medium as recited in Claim 14, wherein the step of creating and storing a Connected Group Space representation further comprises the steps of:

identifying a subnet associated with the source device and the destination device;

determining a set of network links that link one or more network devices in the managed network; and

determining an assignment of ports of network devices.

18. (Original) The computer-readable medium as recited in Claim 14, wherein the step of creating and storing a Connected Group Space representation further comprises the steps of:

identifying all Virtual Local Area Networks (VLANs) associated with a subnet associated with the source device and the destination device; and

50325-0088 (1507)

21. (Original) The computer-readable medium as recited in Claim 20, further comprising the step of creating one Connected Group link for each pair of interfaces within each network device, wherein each interface is associated with the subnet of the source device and the destination device, and is in a forwarding state.

22. (Original) The computer-readable medium as recited in Claim 21, further comprising the step of checking a spanning tree status for each interface within each network device to determine whether the interface is in the forwarding state.

23. (Original) The computer-readable medium as recited in Claim 14, wherein the step of identifying an optimized path in the Connected Group Space representation further comprises the step of finding a shortest path between a Connected Group source node and a Connected Group destination node.

24. (Original) The computer-readable medium as recited in Claim 23, further comprising the step of using a Dijkstra algorithm to find the shortest path between the Connected Group source node and the Connected Group destination node.

25. (Original) The computer-readable medium as recited in Claim 14, wherein the step of transforming the optimized path into the topology space representation further comprises the steps of:

identifying an ordered set of Connected Group nodes associated with the optimized path; and

identifying all Emulated Local Area Networks (ELANs) associated with the subnet associated with the source device and the destination device.

19. (Original) The computer-readable medium as recited in Claim 14, wherein the step of creating and storing a Connected Group Space representation further comprises the steps of:

creating one Connected Group node for any pairs of interfaces across a point-to-point link in the topology space representation;

creating one Connected Group node for any interfaces of the managed network that are directly connected by virtue of being on a same physical medium;

creating one Connected Group node for LAN Emulation interfaces on a same Emulated Local Area Network (ELAN);

creating one Connected Group node for each internal interface of any network device when the network device has an internal interface;

creating one Connected Group node for the source device;

creating one Connected Group node for the destination device; and

creating one Connected Group node for each user interface on any network device when the network device has a user interface.

20. (Original) The computer-readable medium as recited in Claim 19, further comprising the step of determining Connected Group links between Connected Group nodes in a subnet associated with the source device and the destination device.

identifying an ordered set of Connected Group links associated with the ordered set of Connected Group nodes.

26. (Original) The computer-readable medium as recited in Claim 25, further comprising the steps of:

identifying a pair of interfaces associated with each Connected Group link in the ordered set of Connected Group nodes associated with the optimized path; and

generating an ordered set of topology space links from the pairs of interfaces associated with Connected Group links.

27. (Original) A computer data signal embodied in a carrier wave, the computer data signal carrying one or more sequences of instructions for determining a logical path in a managed network between a source device and a destination device at a data link layer, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

creating and storing a Connected Group Space representation of network devices based on a topology space representation of the network devices;

identifying an optimized path in the Connected Group Space representation;

transforming the optimized path into the topology space representation; and

creating and storing the optimized path that was transformed into the topology space representation as the data link layer path.

28. (Original) A computer apparatus comprising:

a processor; and

a memory coupled to the processor, the memory containing one or more sequences of instructions for determining a logical path in a managed network between a source device and a destination device at a data link layer, wherein execution of the one or more sequences of instructions by the processor causes the processor to perform the steps of:

creating and storing a Connected Group Space representation of network devices based on a topology space representation of the network devices;

identifying an optimized path in the Connected Group Space representation;

transforming the optimized path into the topology space representation; and

creating and storing the optimized path that was transformed into the topology space representation as the data link layer path.

29. (New) The method of claim 1, further comprising the step of monitoring network devices by obtaining information about the network devices from information associated with the data linked path.

30. (New) The method of claim 1, further comprising the step of obtaining diagnostic information by obtaining information about the network devices from information associated with the data linked path.

31. (New) The method of claim 1, wherein the data link path is a trace of a path determinable from a bridge forwarding table.

32. (New) The method of claim 1, wherein the data link path is verifiable by comparing information related to the data link path to information from a bridge forwarding table.

33. (New) The computer readable medium of claim 14, wherein the instructions further comprise the step of monitoring network devices by obtaining information about the network devices from information associated with the data linked path.

34. (New) The computer readable medium of claim 14, wherein the instructions further comprise the step of obtaining diagnostic information by obtaining information about the network devices from information associated with the data linked path.

35. (New) The computer readable medium of claim 14, wherein the data link path is a trace of a path determinable from a bridge forwarding table.

36. (New) The computer readable medium of claim 14, wherein the data link path is verifiable by comparing information related to the data link path to information from a bridge forwarding table.

REMARKS

The Examiner is thanked for indicating that claims 6-9 and 19-22 contain allowable subject matter, removing all of the art rejections of the previous Office Action, removing the objection to the drawings, and removing the objection to the specification.

STATUS OF CLAIMS

Claims 1- 36 are pending in the case. Claims 1-28 are original claims, and claims 29-36 are new claims. No claims are amended.

CLAIM REJECTIONS – 35 U.S.C. § 102 and § 103

The Examiner rejected claims 1-4, 10-17 and 23-28 under 35 U.S.C. §102(e) as allegedly anticipated by Hsu (U.S. Patent No. 6,363,318). The Examiner also rejected claims 5 and 8 under 35 U.S.C. §103 as allegedly unpatentable over Hsu in view of Chang (U.S. Patent Application Publication No. 2003/0123448). These rejections are respectfully traversed.

Affidavit under 37 CFR § 1.131

Attached are (1) an Affidavit under 37 CFR § 1.131 and (2) a supporting redacted disclosure document.

The filing date of Hsu, which is August 31, 1999, is less than one year prior to the filing of the present Application, which is March 14, 2000, and therefore does not constitute a statutory bar. The effective date of Hsu as a reference is August 31, 1999. The Affidavit avers the existence, prior to the effective date of Hsu as a reference, of a

working version of subject matter that is an embodiment of the claims. The existence of a working version of the claimed invention constitutes a reduction to practice. Thus, the affidavit in-and-of-itself evidences that the claimed invention was conceived and reduced to practice prior to the effective date of Hsu as a reference.

The disclosure document describes subject matter that is an embodiment of the claimed invention, and in page 8, the disclosure document states

> Cisco Use: This method is currently being used in Cisco PathTool to determine the layer 2 path.

Since the Affidavit avers that the disclosure document was written before the filing date of Hsu, the disclosure document is further evidence that the claimed invention was reduced to practice prior to the filing date of Hsu. Further, the Affidavit avers that the date of the disclosure document is prior to the filing date of Hsu. Thus, the Affidavit, when combined with the disclosure document, is a showing of facts that are of a character and weight as to establish reduction to practice prior to the effective filing date of the reference. Accordingly, Hsu should be removed as a reference, and the rejection under 35 USC § 102(e) and § 103 should be withdrawn.

OBJECTION TO CLAIMS 6-9 AND 19-22

Since the rejection of the base claims should be removed, the objection to claims 6-9 and 19-22 as depending upon rejected base claims should also be withdrawn.

NEW CLAIMS

Each of new claims 29-36 depend on one of claims 1 and 14. Since Hsu should be removed as a reference regarding the base claims of 1 and 14, therefore claims 29-36 are also allowable. Additionally, the passages of Hsu cited by the Examiner never

disclose or suggest (1) using Hsu's method for monitoring, as recited in claims 29 and 33, (2) using Hsu's method for obtaining diagnostic information, as recited in claims 30 and 34, (3) using Hsu's method for tracing a path at level 2, as recited in claims 31 and 35, or (4) that Hsu's method determines a verifiable path that is in a bridge forwarding table, as recited in claims 32 and 36. Further, the title of Hsu is "Constraint-Based Route ***Selection*** Using Biased Cost" (emphasis added), indicating that Hsu uses his method for selecting a route (that presumably was not previous determined), and not for monitoring, obtaining diagnostic information, tracing a path, or determining a path that could have been verified by a comparison with a bridge forwarding table.

DISCLAIMER

The filing of the affidavit and the filing of new claims 28-32, neither affirms nor denies whether the rejections over art of claims 1-5, 10-18, and 23-28 would have been valid were the Affidavit not filed. The Applicant reserves the right to establish differences and/or similarities between Hsu and any of claims 1-5, 10-18, and 23-28.

CONCLUSIONS AND MISCELLANEOUS

The Applicants believe that all issues raised in the Office Action have been addressed and that allowance of the pending claims is appropriate. Entry of the amendments herein and further examination on the merits are respectfully requested.

The Examiner is invited to telephone the undersigned at (408) 414-1213 to discuss any issue that may advance prosecution or any other issues related to this application.

No fee is believed to be due specifically in connection with this Reply. To the extent necessary, Applicants petition for an extension of time under 37 C.F.R. § 1.136. The Commissioner is authorized to charge any fee that may be due in connection with this Reply to our Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Dated: November 26, 2003

David Lewis
Reg. No. 33,101

1600 Willow Street
San Jose, California 95125-5106
Telephone No.: (408) 414-1080
Facsimile No.: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450.

on November 26, 2003 by ______________________
(Date) (Signature)

COPY

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:	**Confirmation Number:** 8997
Mehryar Garakani	**Group Art Unit:** 2697
Serial No.: 09/524,725	**Examiner:** Christopher M Swickhamer.
Filed: March 14, 2000	
For: METHOD OF DETERMINING A DATA LINK IN A MANAGED NETWORK	

DECLARATION OF INVENTOR PURSUANT TO 37 C.F.R. 1.131

Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

Sir:

I, Mehryar Garakani, declare as follows:

1. I am the named inventor of the above-identified patent application and I am familiar with the claimed invention.

2. Prior to August 31, 1999, I had developed a working version of an embodiment of the invention in the United States, and documented a written description of an embodiment of the invention in the United States.

3. I have reviewed the currently pending claims 1-5, 10-18, and 23-28 of the application, and to the best of my recollection, the working version that I developed and the written documentation that I wrote prior to August 19, 1999 is for an embodiment of the invention that is within the scope of claims 1-5, 10-18, and 23-28. Independent claims 1, 14, 27, and 28 are set forth below:

1. A method for determining a logical path in a managed network between a source device and a destination device at a data link layer, the method comprising the computer-implemented steps of:
creating and storing a Connected Group Space representation of network devices based on a topology space representation of the network devices;
identifying an optimized path in the Connected Group Space representation;
transforming the optimized path into the topology space representation; and
creating and storing the optimized path that was transformed into the topology space representation as the data link layer path.

14. A computer-readable medium carrying one or more sequences of instructions for determining a logical path in a managed network between a source device and a destination device at a data link layer, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:
creating and storing a Connected Group Space representation of network devices based on a topology space representation of the network devices;
identifying an optimized path in the Connected Group Space representation;
transforming the optimized path into the topology space representation; and
creating and storing the optimized path that was transformed into the topology space representation as the data link layer path.

27. A computer data signal embodied in a carrier wave, the computer data signal carrying one or more sequences of instructions for determining a logical path in a managed network between a source device and a destination device at a data link layer, wherein execution of the one or more sequences of instructions by one or more processors causes the one or more processors to perform the steps of:

creating and storing a Connected Group Space representation of network devices based on a topology space representation of the network devices;

identifying an optimized path in the Connected Group Space representation;

transforming the optimized path into the topology space representation; and

creating and storing the optimized path that was transformed into the topology space representation as the data link layer path.

28. A computer apparatus comprising:

a processor; and

a memory coupled to the processor, the memory containing one or more sequences of instructions for determining a logical path in a managed network between a source device and a destination device at a data link layer, wherein execution of the one or more sequences of instructions by the processor causes the processor to perform the steps of:

creating and storing a Connected Group Space representation of network devices based on a topology space representation of the network devices;

identifying an optimized path in the Connected Group Space representation;

transforming the optimized path into the topology space representation; and

creating and storing the optimized path that was transformed into the topology space representation as the data link layer path.

4. Attached as Exhibit 1 is a technical document, which is at least part of the written documentation that I wrote, and which additionally evidences that I had a working embodiment of

Patent

the above-identified application, before August 19, 1999. Exhibit 1 is an invention disclosure describing the background, summary of operation, and advantages of an embodiment of the claimed invention. Certain dates and names have been redacted from the document of Exhibit 1, as permitted by applicable laws and rules of the U.S. Patent & Trademark Office; however, Exhibit 1 was written long prior to August 19, 1999.

5. I declare that all statements made herein of our own knowledge are true, and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of title 18 of United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Dated: Nov 11, 2003

By: [signature]
Mehryar Garakani

50325.~~0086 (Seq. No.: 1517)~~
50325-0088 (Seq. No 1507)

COPY

Disclosure Document

CISCO CONFIDENTIAL

This report contains the following 1 ideas

Idea No.	Title	Inventors	Entered	Updated
45781	Method of determining a data link path in a managed network	Mehryar Garakani (mgarakan)		

CISCO CONFIDENTIAL

Method of determining a data link path in a managed network

CPOL No.: 45781	Seq No.: 1507	Status: Pending	Submitted:	Modified:

Idea Details

Inventors:

Mehryar Garakani (mgarakan)	**Phone:** 805 961-3640	**Manager:**	**Dept:** DSP Protocols
Type: Regular	**Division:**SVS	**Site:** --	**Info:** --

Background: Terminology

Layer 2: Refers to layer 2 (data link layer) of ISO networking model

Layer 3: Refers to layer 3 (network layer) of ISO networking model

Vlan: A Virtual LAN consisting of a subset of ports/interfaces on various devices that are part of the same "broadcast domain"

Elan: An Emulated LAN consisting of a subset of ports/interfaces on an ATM fabric that are part of the same "broadcast domain"

Broadcast Domain: A "layer 2" LAN that can consist of multiple virtual LANs, emulated LANs, or physical LAN segments

Subnet: A "layer 3" IP subnet. Typically this is associated with a Broadcast Domain, although the association need not be one to one

NMS: A network management station that collects, analyzes, and displays data on the status of the network and its constituent devices

Bridge: This is used in its more generic sense here. It refers to any device that performs a "layer 2" bridging function from one physical or emulated LAN segment to another. This includes LAN switches, routers having Bridge Groups. For the purposes here it also covers physical layer repeaters (hubs) although technically speaking this is not a common usage for this term.

Topology: A logical map of how each device in a managed network is connected to other neighboring devices. It can be visualized on a diagram that shows the devices and links.

Assume we have a managed IP network that consist of routers, switches, hubs, and other "layer 2" or "layer 3" devices. Furthermore, assume that the following Info items are known about the managed network by the NMS:

Info-1) Topology,
Info-2) Vlan assignment of ports and interfaces,
Info-3) Elan assignment of ports and interfaces,
Info-4) Vlan and elan association with subnets/broadcast domains (i.e., we know which set of vlans and/or elans are part of a given subnet or broadcast domain),
Info-5) Spanning Tree status (FORWARDING, etc.) of each port/interface participating in Spanning Tree Protocol (STP),
Info-6) Bridge Forwarding Table information on LAN switches,
Info-7) End-Station discovery (based on matching ARP Tables entries on routers with Bridge Tables entries on LAN switches) to locate end-station devices (i.e., PCs, workstations, etc.).

The question arises as how can the "layer 2" path for an IP packet involving various devices in this managed network be determined. The "layer 2" path can include multiple "layer 2" LAN switches or hubs (i.e., repeaters), it may also include routers (e.g., router bridge groups).

If we were only interested in the "layer 3" path from one router port to the next, this information could be obtained through the widely available traceroute program in many circumstances (particularly straight forward if the program is run from the source node rather than running from a remote NMS, although doing the later can also be accomplished with the "source routed traceroute" once the first layer 3 hop from source towards destination is determined).

However, the traceroute program is limited in that it can only identify the "layer 3" path. The program works because of the support incorporated into IP protocol for Time To Live (TTL) field and generation of ICMP packets when TTL expires on a packet that has been routed more than the allowed TTL. This allows traceroute program to directly trace the path, by having feedback from devices along the path.

Similar direct tracing support for "layer 2" data link layer protocols across the wide range of technologies, including Ethernet, Lane, Token Ring, EtherChannel, etc. does not exist. Hence the solution needs to be obtained based on the information that can be gathered from the devices by NMS (i.e., those listed in the seven categories above).

Needless to say that this is a significant problem as the ability to determine the layer 2 path in a managed network can be important for network monitoring and diagnostics. This is particularly the case as layer 2 switches are replacing routers in the distribution layer and elsewhere in the networks, and hence layer 3 path traces that jump over from one router port to another can leave a big hole in determining the actual path travelled by a packet from one device to the next.

It should be noted that the layer 2 path fits on top of the Topology map which is assumed known by the NMS. However, knowing Topology itself is not sufficient to know the actual layer 2 path, as there are a typically a large number of solutions that can be overlayed on top of Topology to connect the source device to the destination device.

We can consider the Topology (T) in an abstract sense to be a set of nodes (N) which represent devices and links (L) which join devices together through point-to-point or shared mediums.

To illustrate the problem and or for simplicity we shall make a simplifying assumption regarding Topology initially. Let's make the assumption that the links between devices are point-to-point, e.g., 10BaseT Ethernet, serial links, etc. We will waive this assumption before presenting the solution which would apply to the more generic case.

Each link L is associated with a device port (P1) on one side and a device port (P2) on the other side. We shall see later that ports do not need to be external physical ports, we can consider them to also represent a logical interface or even an internal port when communicating with a device that has an IP address associated with an internal port (e.g., a LAN switch). For now, let's assume all ports are external physical ports.

In this simplified form the problem is to determine the "layer 2" path from some a source port (S) to some destination port (D). The layer 2 path in the above simplified network is the ordered set of all Links (L1, L2, L3, ...) that are traversed from S to D as the packet is going through various bridging devices (such as LAN switches or repeaters). So determining the Layer 2 path (P) is equivalent to determining the ordered set (L1, L2, L3, ...) that is traversed along from one device to the next.

So how can this ordered set be determined? Is this the shortest path on the Topology ? The answer is clearly no. The two devices can be adjacent in the topology (i.e., may be only one hop away in terms of connectivity). However, the actual path the packet takes is constrained by vlan/elan relationships as well as by Spanning Tree configuration.

This could result in a packet to hop along many "layer 2" devices before finally reaching its nearby destination. So although the two devices may be only one hop away in the Topology sense, the actual Level 2 path may consist of many more hops (sort of like traveling from Los Angeles to San Francisco via New York).

This occurs because packet flow is constrained according to the following considerations:

1) Packets would not be bridged on a bridge device from one port to another, unless the two ports are assigned to carry traffic for the same vlan or a translationally related vlan or a binded elan (strictly speaking this is only true as long as the bridge device is performing a standard bridge function and not a routing function or multi-layer switching (MLS). The MLS switches can be considered within the framework presented here, but we will only consider standard switches that do not have MLS

capability in order to provide the simplest presentation of the basic approach.

2) Packets would flow along links that are in Spanning Tree forwarding state and not along links that are blocked. Since the Spanning Tree Protocol generally prefers paths that have a lower cost (e.g., a higher bandwidth), this may not necessarily be the shortest path on the Topology.

The solution presented will be discussed in term of IP networks, although the solution need not be limited to IP and can be applicable to other networking protocols.

Prior Art: ---

Summary: A fact that is noteworthy regarding this problem is that in a functional
network, there is only one unique solution (one l2 path) from source to destination (due to Spanning Tree blocking of redundant paths). This is suggestive of optimization problems, such as minimization or maximization problems.

However, there is a difference in that the solution does not appear as a minimization or maximization problem when looked at in the Topology space. As stated above, the actual path need not be the shortest path in the Topology space (and not the longest path either). This is unfortunate, as there are known algorithms that can be used to find the solution for optimization problems involving a connected network.

The approach that is presented here offers a solution to this problem. It achieves this by transforming the problem from the Topology space into an artificial space called Connected Group (CG) space, whose sole purpose is that it transforms the path finding problem into an optimization problem, whose solution can then be obtained by standard algorithms such as Dijkstra.

Below is provided specific recipe for how such a transformation of the problem can be achieved. The basic procedure can be stated as follows:

1) Based on the information known to NMS, construct a CG space that represents the given path tracing problem,

2) Identify an optimized (shortest path) in the CG space that is an analog (equivalent) of the actual path in the Topology space.

3) Once the solution is found in the CG space, transform the solution back to the Topology space, out of which pops out the ordered set of links (L1, L2, ...) that represents the layer 2 path.

Having stated the problem above in the simplified form by assuming that the links between the devices are point-to-point, now we shall relieve this assumption in order to present a solution that is generic and applies

to point-to-point links as well as when multiple devices are connected through a shared medium.

In this case the topology view is more complex as multiple devices show up connected on the same spot (i.e., the shared medium), however, the path is still an ordered set of links (L1, L2, ...), and our goal is to find it.

The solution presented here covers this more generic case, and is applicable to shared media (e.g. 10Base2) as well as to emulated LANs, e.g. those achieved through ATM LAN Emulation, which can also be thought of as a shared emulated media, as long as we look at the ATM Fabric associated with a particular Elan as a single cloud and not be concerned with the specific cell switching path through the ATM Fabric (i.e., in this case the layer 2 path that is determined would jump over the fabric from one edge device port to another).

The various steps of the procedure above are described below:

1) Constructing the CG space. Just like the topology space the CG space includes a set of nodes (M) that are linked by a set of links (K). However, there are differences in semantics of what a node and a link means in CG space as opposed to the Topology space.

   In CG space, a node can be called a "Connected Group" and is created for any one of the followings:

   a) a pair of ports joined across a point-to-point link on Topology (e.g. a 10BaseT link from a LAN switch to a repeater),

   b) a set of ports that are directly connected, because they sit on a shared physical medium (such as a 10Base2 cable),

   c) a set of Lane interfaces on an elan on an ATM fabric,

   d) an internal port for devices that have internal IP addresses not associated with any of their physical ports (e.g., LAN switches),

   e) a "user port" on a bridge device (i.e., a port that is not connected to any other networking devices, but may have end-stations sitting behind it).

   A node in CG space represents a single port or a connected group of ports (i.e., can be any one of the five items above), unlike a node in the Topology space (which represents a device). The key in understanding a node in CG space is that it is a set of one or more physical, virtual or emulated port/interfaces that are directly connected to other members of the "Connected Group" (if it has more than one member), but for a packet to move from one "Connected Group" to another it must traverse one or more bridging devices.

   The opposite is true for packets going from one port/interface on a Connected Group to another port/interface within the Connected Group. In this case, no bridge devices are needed (this excludes the ATM switches that participate in creating an emulated LAN, as the whole ATM fabric can be viewed as a broadcast entity for each elan).

A question arises as which CG nodes need to be included in CG space. The answer is we need to include a CG node for any item which fits in one of the above 5 categories (a through e) and is associated with the given broadcast domain (or subnet) that we are interested in.

In particular a CG node needs to always be created for source and destination of a layer 2 path. For end-stations the later requires knowledge of which LAN switch port an end-station is behind; this does depend on end-station discovery function of NMS, which was mentioned above (i.e., Info-7). For the purpose of this discussion the CG node associated with the source is refered to as "source CG node" and that of the destination is refered to as "destination CG node". The "source CG node" and "destination CG node" may be one and the same, if the source port and destination ports are on the same Connected Group.

The first step to construct the CG space is to determine the subnet that we are interested in. The subnet can be determined from the "layer 3" trace and by looking at RFC-1213 mib values associated with the "layer 3" interfaces. It was assumed earlier that the NMS can then map this subnet to a broadcast domain, which includes the set of all vlans and elans that are associated with this subnet. The Topology elements that are associated with this set of vlans and/or elans that fit into one of the above five categories can then be used to determine the CG nodes that must be included.

Determining the whole set of CG nodes for a given path tracing problem relies on Info items from NMS which were listed above, in particular Info-1, Info-2, Info-3, Info-4, and Info-7.

Once the CG nodes are determined, the next step is to determine the links between CG nodes. The CG links are all point-to-point (even when Topology includes shared media). Unlike a Topology link, which links two ports on two separate devices, a CG link corresponds with a bridging link that joins two ports on a single bridge device.

CG nodes are joined by point-to-point CG links that are determined as follows. Each bridge device that has ports on the given broadcast domain contributes to a set of CG links. The total set of CG links that are included in the CG space, are the union of all sets from each bridge device that is associated with the broadcast domain.

So assume a bridge device has "m" ports (internal and external) that are associated with the given broadcast domain and are also in the STP forwarding state (i.e., not blocked by Spanning Tree). For each pair of ports on this set of "m" ports a single CG link is added to the CG space. Hence a device that has "m" ports on the broadcast domain would contribute m*(m-1) CG links to the CG space.

Each CG link joins a CG node to another (i.e., is point-to-point). The CG nodes to join are determined based on the association of the bridging ports, that are associated with the CG link, to their corresponding CG nodes.

To determine the CG links, the NMS info items Info-1, Info-2, Info-3,

Info-4, and Info-5 can be used.

Once all CG nodes and CG links are added, the CG space construction is complete and we can proceed to determine the path in the CG space as in step 2 below.

2) Having constructed the CG space the problem of finding the layer 2 path becomes a simple optimization problem, because in CG space the layer 2 path corresponds to the shortest path that joins the "source CG node" with the "destination CG node". Under normal circumstances, when all of the above information items from NMS are known and accurate there will always be a solution and only one solution to this optimization problem. Any other path in CG space that joins the "source CG node" with the "destination CG node" is bound to be longer by one or more hops.

To find the shortest path in CG space the Dijkstra algorithm can be used to identify the ordered set of CG nodes (M1, M2, M3, ..., Mx) traversed from source (M1) to destination (Mx) CG node, along the optimized path.

The ordered Dijkstra set of nodes is the solution in CG space. This needs to be translated back to the solution in the Topology space as discussed below in step 3.

3) The Dijkstra ordered set (M1, M2, M3, ..., Mx) identifies all Connected Groups along the path. Joining each pair of adjacent CG nodes in this path, e.g. (M1, M2) or (M2, M3) is a single Dijkstra link "K?" as was discussed in Step 1 above (the reason there is a unique link is due to the fact that redundant paths are blocked by Spanning Tree Protocol).

So from the ordered set of Dijkstra nodes we can simply generate the ordered set of Dijkstra links (K1, K2, K3, ..., K'x-1'). Each link is associates with a pair of ports on a bridge device as was described above. So once the set of Dijkstra links are known, this can be simply used to generate the set of all ports/interfaces that are traveresed. From the set of these ports/interfaces, we can then immediately generate the set of all links in Topology space (i.e., L1, L2, L3, ...) that constitutes the solution that we seeked.

Assuming that the NMS has consistent information about the network, and the network conditions are normal, the method above should be very effective in finding the layer 2 path. In cases where there is a question about the integrity or consistency of the NMS information, a verification step can be added as below whenever feasible.

Verification Step:

Once we have the solution from the method described above, the Bridge Forwarding Table (Info-6) can be used to verify the solution. This would require the knowledge of destination and source MAC addresses, which can be obtained by querying the router ARP tables (RFC 1213 MIB), as the forwarding tables are keyed off the MAC address.

To do this, for each bridge device along the path which is maintaining

a Bridge Forward Table, a query is made for the destination and source MAC addresses to obtain the incoming and outgoing ports. This is compared against the path that was found above to make sure they match. A mismatch though should be rare would rule out the solution found above (such mismatch can be possible, if NMS view of the network does not reflect reality).

This verification step is not required, but it can provide further assurance that all is well with the solution that has been obtained. The verification step is not always feasible as the Bridge Forwarding Tables age out old information, and unless there has been recent communication between source and destination devices the required information may not be found in the bridge table.

Restatement: ---

Advantages: Provides a simple yet powerful solution to a complex problem.

Cisco Use: This method is currently being used in Cisco PathTool to determine the layer 2 path.

Public Use: None that I am aware of.

Detecting Use: Since path tracing for layer 2 requires a lot of induction based on available information to NMS, etc., generally such tools would provide a window to how they go about achieving a solution, by providing log or trace messages, etc. This is done to give some assurance that the data used and the solution found are valid. Such can be used to detect use by other companies.

If some one is intent to use and hide, only access to source code may do.

Standards: ---

Technologies:

- Network Management
- WAN > Frame Relay

Networking Solutions:

- Large Enterprise > Networking Solutions for Large Enterprise > Switching Solutions for Large Enterprise
- Large Enterprise > Networking Solutions for Large Enterprise > Network Management Solutions for Large Enterprise

Categorization Notes:

8

Categories Summary
[Sw]

Supporting Documents

- ---

Notes ---

CISCO CONFIDENTIAL

A printed version of this page is an uncontrolled document.

HRL LABORATORIES

THIS INVENTION DISCLOSURE IS MA[redacted]
PURSUANT TO MY / OUR INVENTION AGRE[redacted]NT
WITH HRL LABORATORIES, LLC.

SEND COMPLETED DISCLOSURE DIRECT TO:
CORPORATE PATENTS AND LICENSING
Loc. CO, Bldg. C01, M/S A126

SHEET 1 OF 7

b1. TITLE OF INVENTION

Self-forming, Self-maintaining Virtual Backbone-based Radio for Wireless Ad Hoc Networks.

2. INVENTOR(S)

NAME	PAYROLL NO.	SOURCE CODE	LOC	BLDG	MS	PHONE	MANAGER
Bo Ryu	J3254	30-21-40	MA	254	RL96	317-5487	Son Dao
Jason Erickson	60079	23-W1-00	FU	617	C207	732-0185	Dave Derby
James Smallcomb	F8013	23-W1-00	FU	617	C207	732-0514	Dave Derby
Son Dao	E8086	30-21-40	MA	254	RL96	317-5682	Ed Thompson

This is to acknowledge that the above Invention Disclosure has been received by Corporate Patents and Licensing. The disclosure will be reviewed at the next Evaluation Committee Meeting of your organization and you will be promptly informed of the results. If you have any questions please contact the patent attorney listed on the bottom of this form.

This sheet will be returned to the inventor(s) as a confirmation of receipt by Corporate Patents and Licensing.

LOSS OF RIGHTS THROUGH RELEASE TO THE PUBLIC

The right to apply for and obtain a valid patent may be lost as the result of certain activities, such as (1) disclosing the invention outside of the company without an appropriate confidentiality agreement with the receiving party; (2) using the invention publicly; (3) using the invention privately to build or test items that are to be sold publicly; or (4) putting the invention "on sale" by selling or offering for sale an item or product that embodies or uses the invention, or is made or tested by use of the invention. Submitting a proposal with the intent to use the invention in the performance of a resulting contract puts the invention "on sale".

Please inform me immediately of any of these activities or any plans to undertake any of them.

ASSIGNED ATTORNEY: ____________ PHONE ( ) ____________

PATENT DOCKET NO.

[redacted]

[redacted]

H R L LABORATORIES

[redacted]

PD# [illegible]

HRL LABORATORIES

THIS INVENTION DISCLOSURE IS M… PURSUANT TO MY / OUR INVENTION AGR… NT WITH HRL LABORATORIES, LLC.

99W155
RAYTHEON PATENTS & LICENSING RECEIVED
[redacted]

SEND COMPLETED DISCLOSURE DIRECT TO:
CORPORATE PATENTS AND LICENSING
Loc. CO, Bldg., M/S A126

1. TITLE OF INVENTION

Self-forming, Self-maintaining Virtual Backbone-based Radio for Wireless Ad Hoc Networks.

2. INVENTOR(S)

NAME	PAYROLL NO.	SOURCE CODE			LOC	BLDG	MS	PHONE	MANAGER
Bo Ryu	J3254	30	21	40	MA	254	RL96	317-5487	Son Dao
Jason Erickson	60079	23	W1	00	FU	617	C207	732-0185	Dave Derby
James Smallcomb	F8013	23	W1	00	FU	617	C207	732-0514	Dave Derby
Son Dao	E8086	30	21	40	MA	254	RL96	317-5682	Ed Thompson

3. PROOF ON CONCEPTION

A. BY WHOM WAS FIRST DESCRIPTION WRITTEN OR DRAWING MADE?	DATE	TIME SPENT	ACCT. CHARGED	LOCATION OF FIRST DESCRIPTION / DRAWING
Bo Ryu	11/20/98	2 hour	RR8R2NTWL	3H37 (VTC Room)
B. TO WHOM WAS INVENTION FIRST DISCLOSED? Srikanth Krishnamurthy and Yongguang Zhang	DATE ~~11/5/98~~ 11/20/98			

4. REDUCTION TO PRACTICE

A. WAS A DEVICE EMBODYING THE INVENTION CONSTRUCTED AND TESTED OR THE PROCESS PRACTICED?	YES ☐ NO ☐	BY WHOM	DATE STARTED	DATE COMPLETED	TIME SPENT
B. ACCOUNT CHARGED — TIME	ACCOUNT CHARGED — MATERIAL		PRESENT LOCATION OF DEVICE		
C. PRESENT LOCATION OF DOCUMENTS (DATE SIGNED AND WITNESSED), INCLUDING PHOTOS, DRAWINGS, AND DATA SHEETS SHOWING REDUCTION TO PRACTICE					

NOTE: ALL EVIDENCE OF CONCEPTION (FIRST DRAWING AND FIRST WRITTEN DESCRIPTION) AND EVIDENCE OF REDUCTION TO PRACTICE (DEVICE EMBODYING THE INVENTION AND TEST DATA) MUST BE RETAINED.

5. RELATION TO GOVERNMENT CONTRACT

A. DOES THIS INVENTION RELATED TO WORK PERFORMED UNDER A GOVERNMENT CONTRACT?	YES ☐ NO ☐	CONTRACT NUMBER AND TITLE
B. IS INVENTION BEING USED ON A GOVERNMENT CONTRACT?	YES ☐ NO ☐	CONTRACT NUMBER AND TITLE

6. RELATED DOCUMENTS AND DISCLOSURE (BY YOU OR BY ANOTHER). PLEASE ATTACH COPY.

A. IS THERE A PUBLICATION OR PUBLIC PRESENTATION RELATED TO THE INVENTION?	YES ☐ NO ☒	DATE	IDENTIFY
B. ARE THERE ANY RELATED INVENTION DISCLOSURES OR PATENT APPLICATIONS?	YES ☐ NO ☒	DATE	IDENTIFY PD NO. ETC.
C. ARE THERE ANY PROPOSALS OR REPORTS OR OTHER DOCUMENTS RELATING TO THIS INVENTION	YES ☒ NO ☐	DATE Dec 8, 1998	IDENTIFY Technical Specification of Virtual Dynamic Backbone Protocol
D. HAS THE INVENTION BEEN USED, DISCUSSED, DEMONSTRATED OR OTHERWISE DISCLOSED OUTSIDE THE COMPANY (SUCH AS TO A VENDOR OR CUSTOMER)?	YES ☐ NO ☒	DATE	TO / FOR WHOM (COMPANY / PERSON)

HRL PROPRIETARY

PATENT DOCKET NO. LABORATORIES
[redacted]

SIGNATURE INVENTOR [redacted] DATE

SIGNATURE INVENTOR [redacted] DATE

PD# 981213

228C-2 CS MAR 94

HRL LABORATORIES

THIS INVENTION DISCLOSURE IS M… PURSUANT TO MY / OUR INVENTION AGR…ENT WITH HRL LABORATORIES, LLC.

SEND COMPLETED DISCLOSURE DIRECT TO:
CORPORATE PATENTS AND LICENSING
Loc. CO, Bldg., M/S A126

3/17

7. SALE

A. HAS PRODUCT EMBODYING INVENTION OR MADE BY INVENTION BEEN PROPOSED, SOLD, OR OFFERED FOR SALE?	YES ☐ NO ☐	ORDER NO.	ORDER DATE	DELIVERY DATE	DATE OFFERED OR PROPOSED
B. IS PRODUCT EMBODYING INVENTION OR MADE BY INVENTION IN A DELIVERABLE ITEM?	YES ☐ NO ☐	DELIVERY DATE			

99W155
RAYTHEON
… LICENSING
RECEIVED
[redacted]

HRL PROPRIETARY

SIGNATURE INVENTOR [redacted] DATE

SIGNATURE INVENTOR [redacted] DATE

PATENT DOCKET NO.
HRL LABORATORIES
[redacted]
PD# 881813

23C-2 CS MAR 94

HRL LABORATORIES

8. SUMMARY OF THE INVENTION

A. GIVE A BRIEF DESCRIPTION OF YOUR INVENTION, PARTICULARLY POINTING OUT WHAT IS BELIEVED TO BE NOVEL (THE "HEART" OF WHAT IS NEW).

This invention discloses a *novel design of radio capable of efficiently forwarding packets* in an environment called "highly dynamic wireless ad hoc networks": no fixed communication infrastructure exists and radios can freely move around the area. In such an environment, radios capable of performing the following functions are needed:

- To autonomously form and maintain a bandwidth-efficient packet forwarding infrastructure (called *Virtual Dynamic Backbone* or VDB) adapting to high host mobility and RF (radio frequency) connectivity variations.
- To find and forward packets to single or multiple destinations via routes established over VDB.

The disclosed radio design is novel in the following aspects:

1. Virtual Dynamic Backbone (VDB) is created and maintained based on a *light-weight, fully distributed* algorithm that uses *very small broadcast messages* periodically/randomly generated by radios at a fixed rate.
2. The small broadcast messages generated at a fixed rate provide a basis to define "*virtual wire*" in a wireless network.
3. The notion of *virtual wire* makes it possible to accurately quantify *RF connectivity* (or *link quality)* and its stability.
4. Virtual wire also introduces *soft handoff* which allows a radio to connect to a different VDB radio when link quality falls below certain threshold.
5. VDB takes quantified link quality into account while being created/maintained such that when established, the links among radios comprising a VDB are reliable and stable.
6. Message overhead for creating and maintaining a VDB is fixed regardless of radio mobility and link quality variation. This is because the rate of broadcast message generation by each radio is same and fixed.
7. VDB includes sufficient *redundancy*, yielding fast adaptation to network topology change.
8. Packets are forwarded (routed) over *stable and reliable* VDB, increasing the likelihood of successful delivery.

SIGNATURE INVENTOR — DATE [redacted]

SIGNATURE INVENTOR — DATE [redacted]

READ AND UNDERSTOOD BY:

YONGGUANG ZHANG — WITNESS NAME (TYPE) — SIGNATURE — DATE [redacted]

SRIKANTH KRISHNAMURTHY — WITNESS NAME (TYPE) — Srikanth V.K. — SIGNATURE — DATE [redacted]

PATENT DOCKET NO.
HRL LABORATORIES
[redacted]
PD # CS1813

[redacted]

HRL
LABORATORIES

5/9

8. EXPLAIN THE PURPOSE AND ADVANTAGES OF YOUR INVENTION. (WHAT WILL THE INVENTION DO BETTER THAN DONE PREVIOUSLY?)

- Purpose:

1. To support efficient delivery of multicast (one-to-many) and broadcast data.
2. To minimize the impact of radio mobility on route changes/updates.
3. To require no change in the Internet applications running over [TCP/UDP]/IP when used in a constantly changing topology.
4. To efficiently support and manage Quality of Service (QoS) in a wireless ad hoc network
5. To support a large number of radios (on the order of thousands).

- Advantages:

6. Wireless bandwidth is efficiently utilized for multicast/broadcast since packet forwarding is restricted to a small set of radios only (those comprising VDB, or VDB radios).
7. The VDB is created and maintained with sufficient routing redundancy so that it is highly fault-tolerant.
8. The soft handoff algorithm allows radios to react to link quality degradation much more efficiently and quickly.
9. QoS support (bandwidth, delay, and loss) is easier to manage since (a) VDB is designed to be relatively stable and reliable in the face of highly dynamical nature of wireless ad hoc networks), (b) redundancy permits finding alternate route(s) quickly despite topology changes, (c) *virtual wire* makes it possible to detect link quality degradation
10. Internet applications running over Internet Protocol (TCP/UDP/IP) are seamlessly supported in a wireless ad hoc network using this radio.

99W155
RAYTHEON
PATENTS & LICENSING
RECEIVED

[redacted]

HRL PROPRIETARY

SIGNATURE INVENTOR [redacted] DATE

SIGNATURE INVENTOR [redacted] DATE

READ AND UNDERSTOOD BY:

YONGGUANG ZHANG
WITNESS NAME (TYPE) SIGNATURE [redacted] DATE

SRIKANTH KRISHNAMURTHY Srikanth · V.K
WITNESS NAME (TYPE) SIGNATURE [redacted] DATE

PATENT DOCKET NO.

HRL LABORATORIES

[redacted]

PD # 581213

HRL LABORATORIES

8. SUMMARY OF THE INVENTION *(Continued)*

C. IDENTIFY THE COMPANY PROGRAM OR PRODUCT LINE TO WHICH THE INVENTION APPLIES, AND THE EXPECTED VALUE TO THE PROGRAM OR PRODUCT LINE. ALSO IDENTIFY POTENTIAL COMMERCIAL APPLICATION OF THIS INVENTION, INCLUDING AUTOMOTIVE APPLICATIONS, IF ANY.

Raytheon's C3I is developing the next generation Tactical Internet system for large-scale wireless ad hoc networks. This program requires an innovative and practical solution to design a radio that can provide a robust and scalable communication infrastructure suitable for tactical wireless environment while seamlessly supporting Internet applications. The disclosed radio design is a highly potential candidate for this program.

Other business units involved in wireless ad hoc networking systems may benefit from the disclosed invention.

D. IDENTIFY THE PRIOR ART KNOWN TO YOU WHICH IS IMPROVED UPON OR DISPLACED BY YOUR INVENTION, AND STATE IN DETAIL, IF KNOWN, THE DISADVANTAGES OF THE CLOSEST PRIOR ART.

1. Efficient broadcast/multicast support:

Most existing proposals for ad hoc routing in wireless networks have no concept of "backbone" as it usually implies "static" topology. Those proposals support broadcast via "flooding" the network, yielding inefficient utilization of bandwidth. Furthermore, multicast cannot be efficiently supported since there is no common packet forwarding structure similar to "static backbone." One recent proposal, called *Spine*, exploits a self-organizing structure similar to our of *virtual dynamic backbone* (VDB) [1]. The key difference between their Spine and the VDB is *constant overhead, virtual wire*, and *soft handoff*. The major disadvantage of Spine is that it does not consider link quality dynamics; it views a connectivity as on/off which is very unrealistic.

2. Quality of Service Support:

Providing QoS support in a wireless ad hoc network is of increasing interest in many military and commercial applications. VDB provides both redundancy and soft handoff, both of which are used to minimize the change of VDB topology. This prevents ongoing communication flows from suffering from the topological change of VDB, thereby supporting uninterrupted (or minimally interrupted) QoS. Other proposals are forced to update routing information when a topological change occurs, yielding temporary suspension of packet forwarding until the new route is found. The amount of time it takes to find such a new route can be significant.

[1] B. Vadubur et al. "Spine Routing," To appear in *Journal of Clusterin*[redacted]

SIGNATURE INVENTOR — DATE [redacted]

SIGNATURE INVENTOR — DATE [redacted]

READ AND UNDERSTOOD BY:

YONG GUANG ZHANG — WITNESS NAME (TYPE) — SIGNATURE — DATE [redacted]

SRIKANTH KRISHNAMURTHY — WITNESS NAME (TYPE) — Srikanth - V K — SIGNATURE — DATE [redacted]

PATENT DOCKET NO.
HRL LABORATORIES
[redacted]

PD# 981213

HRL
LABORATORIES

SHEET 9 OF 9

9. DETAILED DESCRIPTION

DESCRIBE YOUR INVENTION IN DETAIL, USING NECESSARY ADDITIONAL SHEETS

A. BE SURE THAT EACH SHEET IS DATED, AND SIGNED BY EACH INVENTOR AND TWO WITNESSES. (HAC FORM 236C-6 CS SHOULD BE USED, IF PRACTICAL).

B. ATTACH COPIES OF DRAWINGS OR DETAILED REPORTS HELPFUL IN UNDERSTANDING HOW YOUR INVENTION WORKS

C. IF YOUR INVENTION HAS BEEN TESTED, BRIEFLY SUMMARIZED THE TEST RESULTS WHICH CONFIRM THE FUNCTIONS AND ADVANTAGES LISTED IN 8 B ABOVE.

Please see the attached report entitled "Virtual Dynamic Backbone Protocol (VDBP): Technical Specification" written in Dec. 8rd, 1998. It includes detailed design of radio that performs VDBP, including message formats and unicast/multicast routing. The correctness of the disclosed radio design has been tested using computer simulation under various network topologies and changes. This is an ongoing effort, but preliminary results show that VDBP works as desired under diverse mobility patterns. Detailed test results will be provided as available.

99W155
RAYTHEON
PATENTS & LICENSING
RECEIVED
[redacted]

SIGNATURE INVENTOR — DATE [redacted]

SIGNATURE INVENTOR — DATE [redacted]

READ AND UNDERSTOOD BY:

YONGGUANG ZHANG — WITNESS NAME (TYPE) — SIGNATURE — DATE [redacted]

SRIKANTH KRISHNAMURTHY — WITNESS NAME (TYPE) — Srikanth - V.K — SIGNATURE — DATE [redacted]

PATENT DOCKET NO.
HRL LABORATORIES
[redacted]
PD # 981213

236C-6 CS MAR 94 (EACH PAGE UPON WHICH INFORMATION IS ENTERED SHOULD BE SIGNED AND WITNESSED)

Virtual Dynamic Backbone Protocol (VDBP): Technical Specification

Bo Ryu[*]
Jason Erickson[†]
Jim Smallcomb[†]
Son Dao[*]

[redacted]

1 Introduction

In this invention disclosure, *Virtual Dynamic Backbone Protocol* (VDBP) is described in detail. The VDBP is a light-weight, distributed routing protocol for wireless ad hoc networks with high mobility and link quality variations. In particular, VDBP performs two tasks: (i) creating and maintaining a VDB; and (ii) routing (both unicast and multicast). A VDB is a small set of connected mobile nodes such that when formed, each node is either part of the VDB or a one-hop away from the VDB. For example, in Fig. 1, the set {3,4,7,8,11,12,13,17} comprises a VDB.

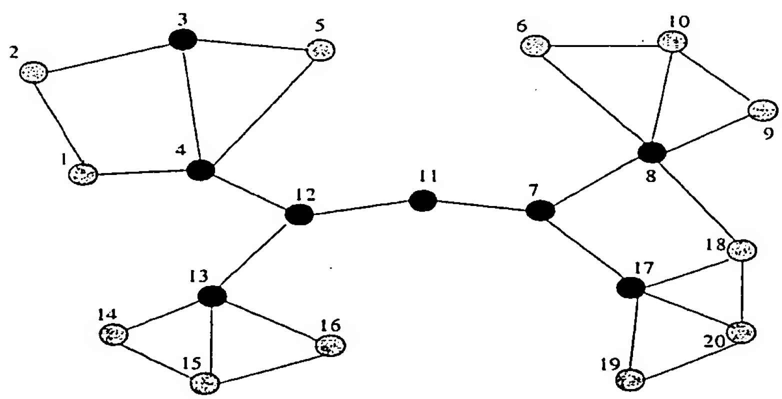


Figure 1: An example of virtual dynamic backbone (VDB).

The basic motivation for the VDBP is to dynamically construct and maintain an optimal packet forwarding structure for the purpose of broadcast packet dissemination. The major benefit of this structure is that it will provide optimal RF channel usage by restricting packet forwarding within only a small subset of the nodes (the VDB nodes). In addition, both unicast and multicast packet routing can be readily achieved using this optimal structure.

From the protocol stack point of view, the VDBP is the upper sublayer of the MAC protocol

[*]HRL
[†]Raytheon

[redacted]

(medium access control protocol) serving the network layer protocol such as IP.

2 Notations

- A flag is used to distinguish whether a node is backbone-capable (Y) or not (N).

- At any time, each node is in one of the three colors: *black* (B) when it becomes part of the VDB (VDB node), *green* (G) when it has at least one black node as its neighbor, and *white* (W) when it has no black node as its neighbor. Only black node forwards packets.

- Each node is assigned a unique 2-byte hardware (MAC) address (node ID).

- $N_1(v)$ denotes the one-hop neighbors of a node v. If two nodes are one-hop neighbors of each other, they are *adjacent*.

- LQ_{ab} represents the link quality from node a to node b observed at b (i.e., outbound link quality from a to b at a, or inbound link quality from a to b at b).

- $\delta(v)$ denotes the number of neighbors of node v with acceptable inbound-only or both inbound and outbound link qualities, referred to as *degree*. When v is white, only inbound link quality with its neighbors is used for calculating $\delta(v)$. When green, depending on the color of its neighbors, either inbound-only (for white and green neighbors) or both inbound and outbound (for black neighbors) link qualities must be acceptable to be counted for degree. When black, both link qualities are used except for white neighbors.

- Let $u \in N_1(v)$, i.e., node u is a neighbor of v. The node v maintains a table called *neighborhood information table* (NIT) which contains $N_1(v)$, color and backbone capability (flag) of each and every node in $N_1(v)$, $\delta(u)$, inbound and outbound link qualities (LQ), expected GBM sequence number, and link quality estimation timer T_{LQE}. The last two columns are used for link quality monitoring described in Sec 5.1.

  For the purpose of illustration, assume that node 3 is B, node 6 is W, and nodes 1,2,4,5 are G. Further, let $N_1(1) = \{2,3\}, N_1(2) = \{1,3,4\}, N_1(3) = \{1,2,4,5\}, N_1(4) = \{2,3,5\}, N_1(5) = \{3,4,6\}, N_1(6) = \{5\}$. Figure 2 illustrates this example, and the node 3 will have the NIT as given in Table 1.

- Each node generates Global Broadcast Messages (GBMs) at a fixed rate. When white, GBMs are generated at constant intervals. When green or black, GBMs are generated at independent and identically distributed random intervals following truncated exponential distribution.

- Each node maintains a routing table based on the received GBMs from nodes farther than one hop. Two routes are maintained for each destination: primary route and secondary route.

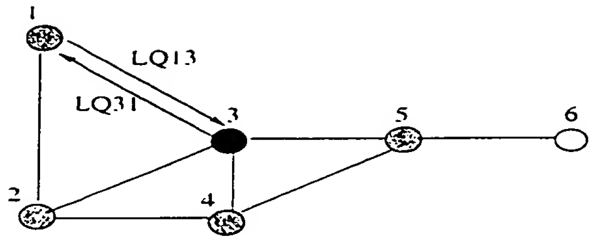


Figure 2: An example of node coloring and two-way link quality notation.

u ($\in N_1(3)$)	Color/Flag	$\delta(u)$	LQ_{u3}	LQ_{3u}	Expected Seq. Nr.	T_{LQE}
1	G/Y	2	15	10	31	26
2	G/Y	3	14	12	327	27
4	G/Y	3	15	16	15	19
5	G/Y	3	10	13	88	5

Table 1: Neighborhood information table (NIT) of node 3 corresponding to Fig. 2.

The secondary route serves as a backup route in case the primary route is not available. Each route is associated with the next-hop gateway, distance metric (currently hop count (HC) only) and *stale timer* (T_{st}). T_{st} represents the remaining time after which this route becomes stale. An example of a routing table at node 4 in Figure 1 is given in Table 2. Routing table

Destination ($\mathcal{D}$)	*Primary*			*Secondary*		
	G	HC	T_{st}	G	HC	T_{st}
1	3	2	25	5	3	21
6	3	2	18	5	3	8

Table 2: Example of routing table at node 4 corresponding to Fig. 1.

management will be described in detail in Section 5.5.

- Each node (say u), black or green, has at least one black neighbor b via which u is attached to VDB. We refer to this black node v as *virtual attachment point* (VAP) of u, or $v = V(u)$. Node u considers v as its VAP if v is a primary gateway. If there are multiple primary gateways, then all of them become VAPs.

- Each green node designates one of its VAPs as *Connection Point (CP)* for the purpose of detecting and connecting disjoint VDBs (see Sec 5.4).

- Each node constantly monitors the neighborhood change and the inbound (received) communication link quality between itself and each of its neighbors by employing the dynamic link quality estimation algorithm described in Sec 5.1.

- Each black node determines whether it needs to remain black as their VAPs by maintaining

VAP timer T_{VAP}. Whenever a black node receives a request to (continue to) become VAP from its neighbor (regardless of its color), it resets T_{VAP} to VAP_TIMEOUT. If no request is received until T_{VAP} expires, the black node changes its color to green after it determines its VAP from its NIT. This VAP request will be specified in the VDBP header (Section 3).

- All the destinations in the routing table and one-hop neighbors of a node i are said to be in *cluster* i, and node i is *cluster center* of cluster i. Each node belonging to a cluster, including cluster center, is called *cluster member*. The instantaneous maximum distance (in hops) between the cluster center and a cluster member is called *radius*. *Cluster size* is defined as the number of cluster members. A fully connected VDB will look like a heavily overlapping clusters linked via cluster centers. This is illustrated in Figure 3. Note that cluster size will

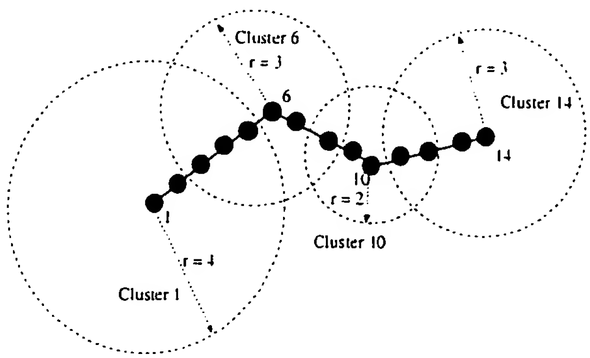


Figure 3: Virtual Dynamic Backbone viewed in terms of *clusters*.

be limited by the maximum size of routing table and NIT. Also, radius of a cluster depends on how clustered nearby nodes are. If nodes are heavily clustered in a region, radius will be likely to be small.

- Separate from routing table, each node maintains a small table called *route cache* which contains route information (gateway and hop count) for destinations who are not in the routing table (since destination is too far away). A node creates a route cache whenever a unicast route query has been successfully made (i.e., route response message has arrived with route information). Therefore, all the nodes involved with setting up a successful inter-cluster route will have corresponding route caches. Each route cache entry is removed when no more packets are generated over this route for a fixed amount of time (ROUTE_CACHE_TIMEOUT). The format of route cache is identical to routing table except that no secondary route exists.

3 VDBP Packet Format

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  -----
|   Destination Node ID         |      Next-Hop Node ID         |   /|\
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |
|    Source Node ID             |    Forwarding Node ID         |    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Hop Count    |      TTL      |           Seq. Number         | 20 Bytes
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Packet Length     |  LQ   | PT | C |Flags| Degree or CS   |    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+    |
|                         Header Option                         |   \|/
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+   ----
|                               .                               |
|                            Payload                            |
|                               .                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Checksum                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A VDBP packet contains 16- or 20-byte header (depending on the presence of header option) as shown above with the following fields:

- *Destination Node ID ($\mathcal{D}$):* The MAC address of the final (ultimate) destination node of the packet. This field is never changed while being forwarded. A subset of this address space will be reserved for supporting multicast. Two special IDs are defined: `GLOBAL_BROADCAST` for all nodes, and `VDB_BROADCAST` for black nodes only.

- *Next-Hop Node ID ($\mathcal{N}$):* This field has three meanings depending on the value of Hop Count field and the colors of the nodes involved with transmitting/receiving this message. First, in all cases, it is the address of the next-hop (receiving) node towards $\mathcal{D}$. If $\mathcal{D}$ is not the one-hop neighbor of the source node, this field will be changed by every forwarding node along the path (route). Second, again in all cases, this field also indicates the node to which link quality update is provided. Finally, except one case, this field also serves as "VAP Request" by the node that originates or forwards this message. The case being excluded from this case is when the forwarding node is black, the next-hop node is green, and the value of Hop Count is greater than 1.

- *Forwarding Node ID ($\mathcal{F}$):* The address of the node by which this packet is forwarded. When a VDBP packet is generated for the first time, $\mathcal{F} \equiv \mathcal{S}$.

- *Source Node ID ($\mathcal{S}$):* The address of the node which originated this packet.

- *Hop Count (HC):* The number of hops that this packet has been forwarded.

- *Time-To-Live (TTL):* The maximum number of hops that this packet can be forwarded.

- *Sequence Number (SN):* a 2-byte number that, in combination with the source node ID, uniquely identifies the packet.

- *Packet Length*: The length of the packet in bytes. The maximum packet length is 4096 bytes.

- *Link Quality Feedback LQ):* Link quality feedback for node $\mathcal{N}$. If $\mathcal{N} = n$ and $\mathcal{F} = f$, this field is taken from LQ_{nf} of the node f's NIT, inbound link quality for neighbor n at f. Once this value is received by node n, it becomes the outbound link quality for its neighbor f.

- *Packet Type (PT):* Type of VDBP packet (3 bits).

PT	VDBP Packet Type
0	Global Broadcast Message (GBM)
1	Route Query Message (RQM)
2	Route Response Message (RRM)
3	Multicast Group Join Message (MJM)
4	Multicast Group Leave Message (MLM)
5	Encapsulated payload is IP packet
6-7	Reserved

Table 3: VDBP packet types and corresponding *PT* values.

- *Color (C):* Color of the node *transmitting* this packet. ("00" - White, "01" - Green, "10" - Black, "11" - Reserved)

- *Flags:* The first bit indicates whether the node transmitting (either generating or forwarding) this message is backbone-capable. The second bit indicates whether the last field is degree or cluster size. The last bit indicates whether header option is present ("0" - N, "1" - Y).

- *Degree (δ) or Cluster Size (CS):* Degree or cluster size of the *forwarding* node depending on whether the second bit of the flags has been set.

- *Header Option:* When the last bit of flags is set, this field includes additional information specific to packet type. An RQM includes IP address in the header option if address resolution from IP address to node ID is needed. Also, GBMs may include critical QoS information such as accumulated delays, packet loss ratio, and available bandwidth.

- *Checksum:* Packet checksum computed over both VDBP header and payload.

4 VDBP Messages

4.1 Global Broadcast Message (GBM)

GBM acts as a "signaling message" for VDBP, and is an empty VDBP packet with $PT = 0$. Each node computes its inbound link quality for each of its neighbors based on the dynamic link quality estimation algorithm described in Sec 5.1. Once a node turns green or black, it maintains outbound link quality with each of its VAPs based on the LQ update provided by its respective VAP.

The GBMs are *generated* under the following rules with $HC = 1$:

- A white node generates a GBM every `GBM_INTERVAL_WHITE` seconds with $\mathcal{F} = \mathcal{S}$ and $\mathcal{D} = \mathcal{N} =$ `GLOBAL_BROADCAST`.
- Once a white node detects the black neighbor b qualified as its VAP, $\mathcal{N} = b$; otherwise, $\mathcal{N} =$ `GLOBAL_BROADCAST`.
- When green or black, the node generates GBMs according to exponentially distributed GBM inter-arrival times with mean `GBM_INTERVAL_AVG` and maximum `GBM_INTERVAL_MAX` sec.
- A green/black node specifies $\mathcal{N}$ from its VAP_{list} in a round-robin manner.
- The TTL value is always given by $r + 1$ where r is the radius just before generating a new GBM. This permits gradual increase in cluster size. If the routing table is empty, then $r = 1$, yielding $TTL = 2$.
- White and green nodes do not transmit their cluster sizes; only black nodes alternatively transmit degree and cluster size i the header of each and every GBM being generated and forwarded.

Once a node receives a GBM, it is processed and forwarded (by black nodes only) as follows:

- Each and every node updates its routing table.
- Neither white node nor green node forwards GBMs. It only updates its NIT.[3]
- During the Backbone Selection Process (described later), upon receiving a GBM a white node updates NIT such as color and degree changes, and resets the backbone selection timer T_{BSP} to `BACKBONE_SELECTION_TIMEOUT`.

[3] We note that those green nodes which have just transitioned from black may forward GBMs in its buffer for a small period of time. This will be described later.

- Each and every node updates inbound link quality for all received GBMs as described in Sec 5.1.

- When a node a receives a GBM with $\mathcal{N} = a$ and $\mathcal{F} = b$, a updates the outbound link quality LQ_{ab} from the header of the received GBM.

- When a black node a *forwards* a GBM with $\mathcal{N} = b$ where b is a neighbor of a, it fills the LQ update field with LQ_{ba}. In other words, a provides the outbound link quality LQ_{ba} with respect to b. b is chosen from its NIT in a round-robin manner.

- When a green node g receives a GBM from its neighbor n (regardless of its color) with $\mathcal{N} = g$ and $HC = 1$ (implying that n is requesting g to become its VAP), g checks if the inbound link quality LQ_{ng} exceeds the predefined link quality threshold `LQ_THRESHOLD`). If so, it changes to black and starts forwarding GBMs; otherwise, it remains green.

- Upon receiving a GBM, a black node decreases TTL by one. If TTL is 0, it simply discards the message after updating NIT and routing table accordingly.

- Just before forwarding any GBM, a black node increases HC by one and forwards with $\mathcal{N} = n$ where n is the one-hop neighbor to which the link quality update is provided. The node n is chosen from the NIT in a round-robin manner.

- A node n receiving a GBM from its VAP b with $\mathcal{N} = n$ and $\mathcal{F} = b$ creates or updates its LQ_{nb} and LQ_{bn}.

4.2 Route Query Message (RQM)

RQM is used to find a route to the destination when it is outside the cluster of the source and no route cache exists. Depending on whether the source also needs address resolution (from IP address to node ID), an RQM takes a different packet format. When address resolution is needed (in addition to route query), the header option field is filled with the IP address of the destination. Otherwise (the source node already knows the node ID of the destination node), only a two-byte payload of the destination node ID is included with no header option. RQM is generated with $\mathcal{D}$ = `GLOBAL_BROADCAST` if address resolution is needed or `VDB_BROADCAST` otherwise. For both cases, TTL = `MAX_TTL`.

4.3 Route Response Message (RRM)

Similar to GBM, RRM has no payload. It includes IP address in the header option field only if the corresponding RQM included address resolution request. Unlike RQM which is sent as a broadcast message, RRM is generated as a unicast packet since the destination of the RRM is the source of

RQM. TTL is given by HC of the received RQM plus a margin (for taking topology change into account during route discovery).

4.4 Multicast Group Join Message (MJM)

MJM also has no payload, and is generated with $\mathcal{D}$ = *group address* (G) and $\mathcal{N} = \mathcal{CP}$. Depending on the status of its multicast routing table, CP and other black nodes decide whether to forward this join message or not with appropriate $\mathcal{N}$. Details will be given in Sec 5.7

4.5 Multicast Group Leave Message (MLM)

MLM takes the same packet format as its counterpart, MJM.

5 VDBP Logics

5.1 Link Quality Estimation (LQE)

The LQE algorithm is designed to accurately monitor changes in link quality between two adjacent nodes (say n_1 and n_2). Fast and accurate detection of link quality change (both increase and decrease) is crucial for maintaining a stable and reliable VDB under extremely dynamic wireless link characteristics.

The ability to successfully exchange packets (GBMs) with neighboring nodes is the basis for LQE algorithm. Before describing the algorithm in detail, we note that the rate of GBMs received from n_2 at n_1 varies dramatically depending on the color of n_2. Suppose each node generates GBMs at an average rate of R pkts/sec, and an average cluster size is CS. If n_2 is green, then n_1 will receive GBMs from n_2 at the rate of R. But if n_2 is black, then n_1 will receive GBMs approximately at the rate of $CS * R$ since it will forward GBMs generated from all nodes within n_2's cluster. In other words, a black neighbor transmits GBMs at a much higher rate than a green neighbor. For this reason, we propose two different LQE techniques:

- Loss-based LQE for green/white neighbors (LB-LQE)
- Rate-based LQE for black neighbors (RB-LQE)

Both techniques are used to monitor and estimate *inbound* link quality for all one-hop neighbors. Since outbound link quality for n_2 at n_1 is identical to inbound link quality for n_1 at n_2 ($LQ_{n_1 n_2}$), and every node piggybacks this information in the VDBP header, we need to estimate inbound link

quality only. We note that outbound link quality is maintained between green-black and black-black pairs only.

Whenever n_1 determines (from either LB-LQE or RB-LQE) that the inbound link quality for its neighbor n_2 has increased (decreased), it increments (decrements) $LQ_{n_2 n_1}$ by LQ_INC (LQ_DEC). Increment and decrement values (LQ_INC and LQ_DEC) are set such that link quality decrease is more pronounced than increase. The link quality varies between 0 and LQ_MAX. A link is considered "good" if it is equal to or exceeds LQ_THRESHOLD. When a new neighbor is detected, its link quality is initialized to LQ_INIT.

5.1.1 LB-LQE

Each time a new GBM is received from its non-black neighbor, the following actions are performed:

1. Compute the number of GBMs lost by comparing the expected sequence number and the new sequence number. Denote it as L_{GBM}.
2. If $L_{GBM} = 0$, then increment the inbound link quality by LQ_INC (if it has not reached LQ_MAX) and denote the resulting value as LQ.
3. Compute $C = \lceil LQ/\text{LQ\_DEC} \rceil$, the factor that determines the period after which this neighbor will be no more considered as neighbor.
4. Reset the corresponding T_{LB-LQE} to $C * I$ where $I =$ GBM_INTERVAL_WHITE for a white neighbor or GBM_INTERVAL_MAX for a green/black neighbor.
5. On the other hand, if $L_{GBM} > 0$, then decrement the inbound link quality by $(L_{GBM} - 1)*$ LQ_DEC, and denote the result as LQ.
6. If $LQ \leq 0$, remove this neighbor from NIT. Otherwise, perform 3 and 4.

The above rules ensure that the value of T_{LB-LQE} is dynamically adjusted reflecting the current link quality.

5.1.2 RB-LQE

The rate-based LQE applies to black neighbors only, and is based on the fact that a black node b will transmit GBMs at the rate that fluctuates around $CS(b) * R$ where R is GBM generation rate at each non-white node (1/ GBM_INTERVAL_AVG), and $CS(b)$ represents the current cluster size of b. A node n estimates the rate of GBMs received from b over a fixed period of time (RB_LQE_INTERVAL

sec). If this rate is below a threshold, then the inbound link quality is decremented by LQ_DEC. Otherwise, it is incremented by LQ_INC. The rate threshold value for b, $R_{th}(b)$, is given by

$$R_{th}(b) = \alpha * CS(b) * R$$

where $0 < \alpha < 1$. If α is 0.75, then the link is considered degraded if the estimated rate is smaller than 75% of the expected rate. The node estimating the rate updates $CS(b)$ from the received GBM header.

5.1.3 Constants and Default Values

Constant	Default Value
LQ_INC	1
LQ_DEC	3
LQ_MAX	15
LQ_INIT	7
LQ_THRESHOLD	9
RB_LQE_INTERVAL	1 sec
α	0.75

Table 4: Default values for constants used for LQE.

5.2 Backbone Selection Process (BSP)

The BSP runs only when a node is in white color and does not have green or black neighbors in its NIT. When a node turns on its power, its color is white and builds its NIT based on the received GBMs from its neighbors. Each time it receives a GBM, it updates its NIT and resets T_{BSP} if a change in NIT occurs (in terms of color and/or degree). The timer T_{BSP} is used to trigger Backbone Selection Process for white nodes. As soon as T_{BSP} expires, a white node decides whether it is eligible to be part of VDB by comparing its degree with that of its one-hop neighbors.[4] If its degree is highest, it changes its color to black and starts forwarding GBMs. Minimum ID is used to break tie between neighbors with the highest δ of equal value.[5] Otherwise, it remains white and resets T_{BSP}.

If a white node w detects a black neighbor b (either a new neighbor or color change of an existing neighbor) with $(LQ_{bw}, LQ_{wb}) \geq$ LQ_THRESHOLD before T_{BSP} expires, w changes its color to green and starts sending its GBMs with $\mathcal{N} = b$. Otherwise, it remains white.

If a white node is not elected as a black node nor does it hear from any black node during BACKBONE_EXPANSION_TIMEOUT secs since it became white, it means that either there is no black

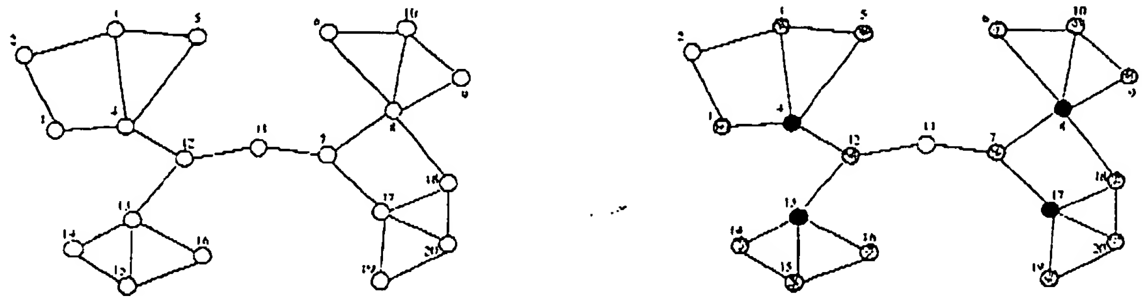
[4] T_{BSP} is not used once it changes to green or black.

[5] Note that there will be no adjacent black nodes after this step.

[redacted]

node in its neighborhood or neighborhood change has been too frequent to trigger BSP. In this case, the BEP is performed as follows.

We illustrate the BSP in Figure 4.



(a) (b)

Figure 4: Illustration of Backbone Selection Process. (a) every node is white; (b) after neighborhood discovery and backbone selection processes, nodes 4, 8, 13, and 17 become black.

5.3 Backbone Expansion Process (BEP)

After a white node w waits BACKBONE_EXPANSION_TIMEOUT secs since it became white, it chooses a green neighbor g with the highest degree (if there are multiple green nodes in its NIT) and that also has an inbound link quality LQ_{gw}[6] exceeding LQ_THRESHOLD, and requests g to become its VAP by specifying $\mathcal{N} = g$. The minimum ID is used to break tie when multiple green neighbors with the identical degree and link quality exist. The green node g that receives a GBM with $\mathcal{N} = g$ from its neighbor w treats this as a request to become a black node. If its LQ_{wg} exceeds LQ_THRESHOLD, g changes to black and starts forwarding GBMs. Once w detects g change its color to black, it changes to green and marks g as its VAP.

If no green neighbor is found during BACKBONE_EXPANSION_TIMEOUT secs, the white node simply waits another BACKBONE_EXPANSION_TIMEOUT secs during which the first green or black neighbor may appear, or a new white neighbor comes up in which case T_{BSP} will be reset so that the BSP can be triggered.

Figure 5 illustrates how BEP works after BSP is done as in Fig. 4-(b).

5.4 Backbone Connection Process (BCP)

The set of black nodes resulting from the previous processes (BSP and BEP) may not be connected. For example, there may be disjoint groups of backbone nodes where only nodes inside the group

[6] No outbound link quality between these two nodes exists since green nodes do not forward GBMs.

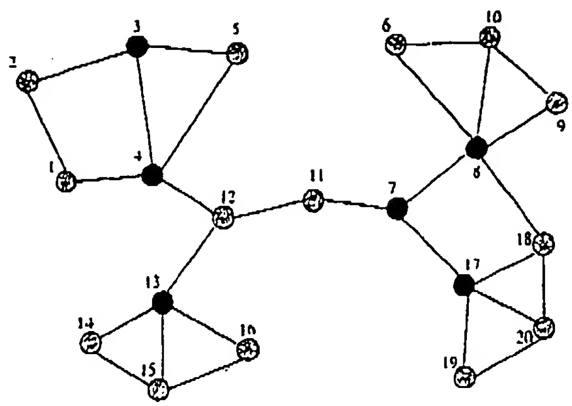


Figure 5: Illustration of Backbone Expansion Process. Nodes 2 and 11 ask nodes 3 and 7 to become black nodes.

are connected. The BCP is employed to connect these disjoint VDBs. It can be easily proven that the above processes will guarantee that at most two green nodes will exist between two disjoint backbone nodes. This means that each green node is responsible for detecting whether it needs to become black to connect disjoint groups of backbone nodes.

The main idea behind the BCP is that if VDB is connected, a green node g will hear two identical GBMs (in terms of source ID and sequence number) from each of its neighbors (except its *Connection Point*, or CP): *direct* GBM, and *forwarded* GBM via its CP.[7] Once a node becomes green, it checks if it has at least two non-white neighbors. If so, it begins to closely watch all the GBMs *originating* from all of its *one-hop* neighbors (except its CP). If g detects that more than a fixed number of duplicate GBMs have not been received for a neighbor n via CP, g is eligible to become black by connecting n and its CP.

Note that each GBM carries a sequence number field. The green node maintains a sequence number table (SNT) whose entry consists of two fields: neighbor node ID, and the sequence numbers of GBMs for which no duplicates have been received (yet) from this neighbor. When the green node receives a GBM, it first checks if the sequence number of this GBM exists in its corresponding SNT entry. If so, indicating that it is a duplicate, the green node discards from the SNT all the sequence numbers that are equal to or smaller than the current one. Otherwise, indicating that it is either a new one or it has been previously discarded, the green node adds this sequence number to the corresponding SNT entry. If it accumulates more than the fixed number of sequence numbers (`MAX_BCP`), the green node is eligible to be part of the VDB by connecting this neighbor and its CP.

Suppose the green node g decides to connect its connection point c and a *black* neighbor b. If b

[7] If g has more than one VAP, it may hear more than two identical GBMs from the same neighbor. Those duplicate GBMs received via VAP(s) other than CP are ignored.

has been one of the VAPs for g, g will have up-to-date two-way link quality with both nodes. If all the link quality values exceed LQ_THRESHOLD, g turns black and starts forwarding GBMs. If no outbound link quality is available for b but LQ_{bg} > LQ_THRESHOLD, then similar to BSP and BEP, g generates a GBM with $\mathcal{N} = b$ and LQ_{bg}. Upon detecting that receiving the forwarded GBM by b with LQ_{gb}, g turns black if the link quality condition is met. Otherwise, it flushes the SNT entry for b, and starts again.

Suppose now that green node g decides to connect its connection point c and another *green* neighbor m. Since no two-way link quality has been established between g and m, g requests m to be its VAP only if LQ_{mg} > LQ_THRESHOLD. Upon receiving this, m also checks LQ_{mg} and decides whether to turn black or not. If m turns black, g will also turn black. Once g turns black as a result of BCP for connecting its CP and neighbor n, n will be added into g's VAP_{list} if n becomes a primary gateway of existing or new routes.

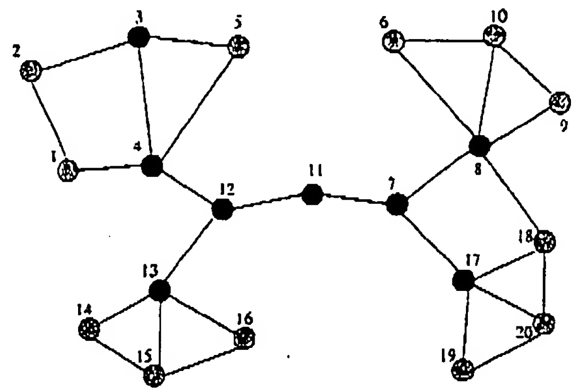


Figure 6: Illustration of Backbone Connection Process. Node 11 (or 12 whichever detects first) asks node 12 to become black to connect nodes 7 and 12. It is possible that nodes 5 and 6 become also black, in which case they are redundant. This redundancy is not necessarily bad since they provide better routes for some nodes, such as between nodes 3 and 10.

5.5 Routing Table Maintenance

Each and every node maintains a routing table (in Table 2) based on the received GBMs. Currently, Hop Count is the only cost metric for routing subject to link quality between gateway and sender. In the future, other metrics such as delay or bandwidth will be taken into account when determining routes.

Suppose node n has received a GBM with $(S, \mathcal{F}, HC) = (s, f, h)$. The following basic rules are then applied for routing table maintenance:

- If there is no route for destination s, a primary route is created for s with f as the primary

[redacted]

gateway and distance h. The corresponding secondary route is empty.

- If a primary route for s exists, but not the secondary route, and f is different from the primary gateway, a secondary route is created for s.
- If both primary and secondary routes for s exist, and the received GBM contains a new gateway f with h better than the secondary but larger than primary route, the secondary route is replaced with f and h.
- If both primary and secondary routes for s exist, and the received GBM contains a new gateway f with h better than the primary route, the primary route is replaced with f and h, and the now-old primary route replaces the secondary route, including the stale timer.
- If both primary and secondary routes for s exist, and the received GBM does not contain a new gateway but has better hop count, only the hop count of the corresponding route is updated.
- For all route updates, corresponding stale timer T_{ST} is set to ROUTE_STALE_TIMEOUT (except when the old primary route becomes the secondary route).
- When a VAP is removed from NIT, all the corresponding routes with this VAP as gateway (either primary or secondary) are also removed. If the route being removed is primary, then the corresponding secondary route becomes primary.
- When routing table is full and a new destination d is detected, the hop count h for d is compared against the radius r. If $h \geq r$, this new destination is ignored. Otherwise, one of destinations with $HC = r$ is replaced with d. If multiple destinations have the same $HC = r$, the destination with its T_{ST} closest to expiration is removed.

5.6 Unicast Routing

When the destination of a unicast packet is either its one-hop neighbor or a cluster member, the source simply sends the packet directly to the destination or via the corresponding primary gateway. However, if the destination is farther than the radius of the cluster, the source must broadcast the Route Query Message (RQM) over VDB.

Any black node receiving an RQM keeps a record of ($\mathcal{S}$, $\mathcal{F}$, $\mathcal{I}$) for a fixed amount of time ($\mathcal{I}$ is either IP address or node ID of the destination). If no corresponding Route Response Message (RRM) is received during that time, this record is thrown away.

When RQM includes an address resolution request, it is the destination node that replies to this query by generating RRM. Otherwise, any black node that knows how to reach the destination

(either the destination is one of its cluster members or route cache is available) generates RRM. Another words, if an RQM is received with $(\mathcal{S},\ \mathcal{F}) = (s,\ f)$, the corresponding RRM is generated with $(\mathcal{D},\ \mathcal{N}) = (s,\ f)$. Upon receiving this message, each black node along the reverse path configures the route cache as follows. Suppose .

The first black node which has the destination within its cluster returns then Route Response Message (RRM). All the intermediate black nodes receiving RRM configures the route and stores in its Route Cache.

5.7 Multicast

With VDB, multicast support is readily achieved since VDB can be used as multicast backbone. Any multicast tree constructed is a subset of VDB. There are two components for multicast support over VDB: (i) group membership management, and (ii) routing.

In IP, local group management is done by Internet Group Membership Protocol. We use a similar technique for our purpose. A multicast group is identified as a unique multicast address. For each multicast group, a black node maintains local group membership information based on two special messages: group join (MJM) and leave messages (MLM).

We adopt the "broadcast-and-prune" approach of Distance Vector Multicast Routing Protocol (DVMRP) of Internet to achieve multicast support over VDB. Each black node maintains multicast routing table consisting of four fields: group address, local members, upstream multicast VAP (VAP_{up}), and downstream multicast VAP(VAP_{down}). Note that the notion of "up" and "down" is with respect to the node that generates the first MJM, regardless of whether it is a source or receiver of the group. This implies that there is only one VAP_{up} but may be multiple VAP_{down}s for a given group. However, our multicast routing algorithm is designed such that any member can act as a source.

1. When a node wishes to join a group G, it generates MJM with $(\mathcal{D},\mathcal{N}) = (G, CP)$.
2. When a black node b receives an MJM for a group D from its neighbor for which it serves as CP, b updates its local membership table. Depending on the multicast routing table status, b decides whether to forward this message further as summarized in Table 5.
3. Any black node b_2 receiving a *forwarded* MJM by a neighbor black node b_1 updates its multicast routing table described in Table 6.
4. When a node n leaves a group D, it generates MLM to its CP b via which it has joined G.[8] Upon receiving this message, b deletes n from its local membership table. If n is the last local

[8] The impact of mobility or link quality fluctuation will be explained while describing Backbone Maintenance.

Status	*Actions*
None of VAP_{up} and VAP_{down} configured	Forward with $\mathcal{N}$ = VDB_BROADCAST if first MJM for G. Otherwise, do not forward.
Only VAP_{down} configured	Do not forward
Only VAP_{up} configured	Forward with $\mathcal{N}$ = VDB_BROADCAST if first MJM for G. Otherwise, do not forward.
Both VAP_{up} and VAP_{down} configured	Do not forward.

Table 5: MJM forwarding rules at CP.

Status	*Actions*
None of VAP_{up} and VAP_{down} configured	$VAP_{up} = b_1$ and forward with $\mathcal{N}$ = VDB_BROADCAST if no local members. Otherwise, $VAP_{down} = b_1$ and do not forward.
Only VAP_{up} configured	$VAP_{down} = b_1$ and forward with $\mathcal{N} = VAP_{up}$ if no local members. Otherwise, $VAP_{down} = b_1$ and do not forward.
Only VAP_{down}(s) present	Add b_1 to VAP_{down} list if new. Do not forward.
Both VAP_{up} and VAP_{down} present	Add b_1 to VAP_{down} list if new. Do not forward.

Table 6: MJM forwarding rules at other black nodes.

member for D and b is at the end of the tree (i.e, only one multicast VAP is configured), it prunes itself from the tree by forwarding the leave message to that multicast VAP and delete it from the multicast routing table. Otherwise (either there are still local members or more than one multicast VAP present), it does not forward the MLM.

5. If a black node b_1 receives a *forwarded* MLM from one of its multicast VAPs, say b_2, it removes b_2 from its multicast routing table. If b_1 has local members for G or there are remaining multicast VAPs (either up or down), the received MLM is not forwarded. Otherwise, b_1 forwards it to other remaining multicast VAPs.

6. With the information of local membership, upstream gateway, and downstream gateway(s) maintained as described above, a multicast routing is done by simply forwarding multicast packets to appropriate multicast VAPs.

We illustrate these rules in Figure 7. Suppose node 1 is the first member to join group G. The join message MJM initiated by node 1 is broadcast over VDB, and all the black nodes except node 3 configure their VAP_{up} accordingly [Fig. 7-(a)]. Now, node 6 joins G via node 7. Since node 7 already has $VAP_{up} = 5$ configured for G, it sends this join message only to node 5, which in turn forwards it to its VAP_{up} 3 [Fig. 7-(b)]. Node 10 now joins as the third member of the group. Node 9 forwards this join message to its $VAP_{up} = 7$, which configures node 9 as its VAP_{down}. This time, node 7 does not need to forward this join message further since it already did when its local member (node 6)

[redacted]

joined [Fig. 7-(c)]. Figure 7-(d) shows how a multicast datagram is forwarded over VDB with node 1 as a source. Each intermediate node in the tree receiving multicast packets from one direction forwards them to the other direction. If there are multiple VAP_{down}s and multicast packets arrive via one of them, they are forwarded to both VAP_{up} and all other VAP_{down}s. Nodes that have only one multicast VAP configured (edges of the tree) do not forward multicast datagrams.[9] When node 1 leaves G, it sends the MLM to node 3 [Fig. 7-(d)]. Since node 1 is the last local member and node 3 knows it is at the edge of the tree, node 3 no longer needs to be part of the tree. Hence, it forwards the leave message to its downstream VAP (node 5) and deletes it from the table. Similar to node 3, node 5 removes itself from the tree by forwarding the received MLM to node 7 and delete 7 from the table. Node 7 unassigns node 5 as its upstream VAP, but does not forward this leave message further since it has a local member. After this step, the multicast tree for G consists of nodes 7 and 9 only. Finally, when nodes 6 and 10 leave the group, all the multicast routing tables of nodes 7 and 9 become empty. The multicast routing table for node 8 becomes empty after timeout since it has not been part of the tree.

5.8 Backbone Maintenance Process (BMP)

As nodes freely move, join (power-on), or leave (power-off) the network, it is very likely that the size of VDB becomes sub-optimal (i.e., too many black nodes compared to the entire network size, leading to high redundancy and unnecessary bandwidth consumption due to GBM forwarding). For example, some black nodes may not be qualified to be part of the VDB any more, or some green nodes are more qualified to become black than their VAPs. Since it is crucial to keep the size of VDB very small to reduce the overhead of GBM forwarding, the BMP must enable unnecessary black nodes to be detached from VDB, and qualified green nodes to join VDB. This will be achieved by ensuring that VDB consists of nodes with *high* degree. Another goal of BMP is to prevent dramatic changes in the VDB structure. If a color or degree change of a node triggers too many route or color changes of nearby nodes, it will undermine benefits of having VDB since packets in transit may get dropped or it may lead to longer routes or packet drops.

We first consider conditions that keep the size of VDB small:

- If a green node's VAP is deleted from its NIT, it is also removed from the VAP_{list}. When VAP_{list} becomes empty, the green node chooses a black neighbor (or green if none) with the highest degree as its new VAP, subject to LQ constraint.
- If no neighbors depend on a black node b as their VAP, the VAP timer T_{VAP} will expire. Then, b chooses one of its black (or green if none) neighbors with highest degree and acceptable two-way LQ as its VAP. Minimum node ID is used as tie breaker.

[9] Note that both nodes 6 and 7 can also transmit multicast datagrams over the same tree.

[redacted]

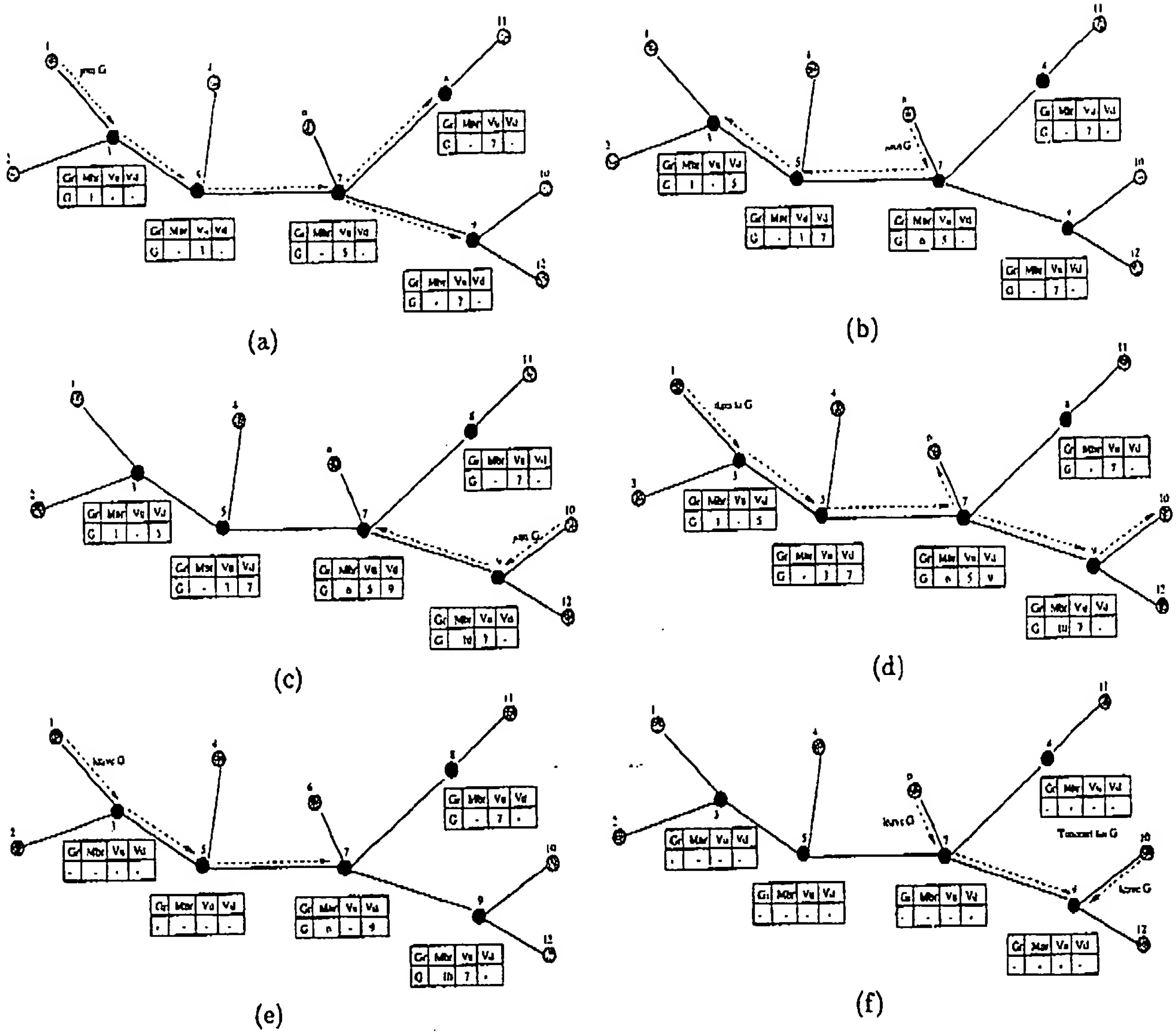


Figure 7: An example of setting up and tearing down a multicast tree over VDB (Gr: group address, Mbr: local members, V_u: upstream VAP, V_d: downstream VAP).

- If a green node g detects either a new green or black neighbor who has higher degree than its current VAP(s), g asks this neighbor to become its VAP subject to two-way LQ constraint. This is likely to trigger many routing updates in g's routing table, and may result in eliminating one or more current VAPs.

Second, in order to keep the structure of VDB stable under high node mobility and link quality fluctuation, we introduce the concept of "soft" handoff. The soft handoff is considered between a green/black node and each of its VAPs only.

When a node n detects that the inbound link quality for its VAP drops below LQ_THRESHOLD, n performs the soft handoff as follows:

- Find all the destinations which have this VAP as their primary gateway.
- Replace the primary routes of those destination with secondary ones as long as secondary gateways satisfy the inbound link quality condition.
- Remove those destinations which have no secondary routes.
- If this was the only VAP, choose a green neighbor with the highest degree as its new VAP subject to inbound link quality condition.

5.9 State Transition Diagram

A node changes its color (state) as shown in Figure 8.

5.10 Timers, Constants, and Default Values

Tables 7 and 8 summarize all the timers and constants used in the VDBP.

5.11 VDBP Pseudo Code

All the algorithms described above can be implemented as follows.

WHITE

1. *Beginning of State at node m:*

   – generate a GBM and set T_{GBM} to GBM_INTERVAL_WHITE;
   – set T_{EXP} to BACKBONE_EXPANSION_TIMEOUT;

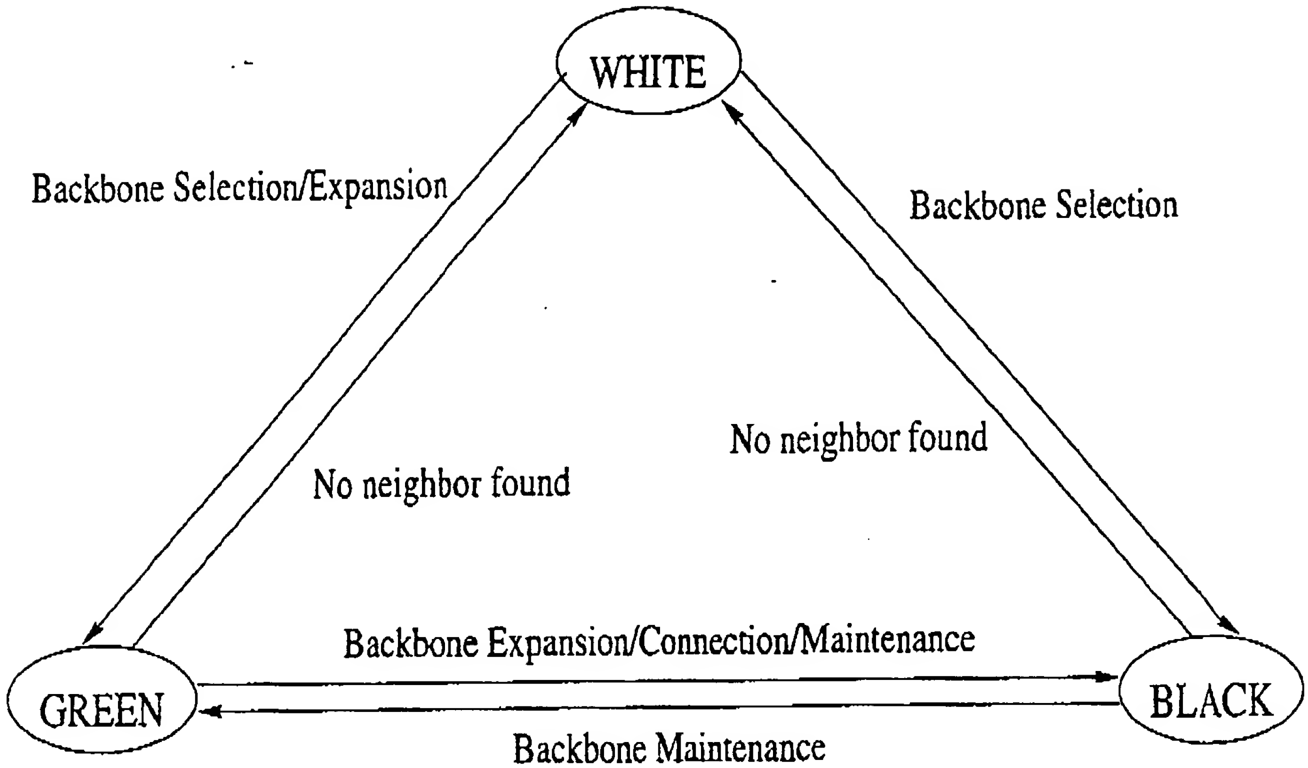


Figure 8: State Transition Diagram of VDBP.

Timer	*Description*	*Relevant Constant (suggested value)*
T_{GBM}	Timer for generating GBM	GBM_INTERVAL_WHITE (3 sec)
		GBM_INTERVAL_AVG (12 sec)
T_{VAP}	Timer for implicit VAP de-registration	VAP_TIMEOUT (30 sec)
T_{BSP}	Timer for BSP	BACKBONE_SELECTION_TIMEOUT (30 sec)
T_{BEP}	Timer for starting BEP	BACKBONE_EXPANSION_TIMEOUT (2 min)
T_{LB-LQE}	Timer for loss-based LQE	dynamically adjusted
T_{RB-LQE}	Timer for rate-based LQE	RB_LQE_TIMEOUT (1 sec)
T_{STBY}	Timer for confirming good LQ with the chosen VAP	STBY_TIMEOUT (3 sec)

Table 7: Timers and related constants.

Constant	*Description (suggested value)*
TTL_GBM_MAX	maximum TTL value for GBM (16)
GBM_INTERVAL_MAX	maximum GBM interval (2 * GBM_INTERVAL_AVG)

Table 8: Other constants.

2. *Received GBM from one-hop neighbor n:*

- if (n is white)
  - o if (new neighbor)
    - – add to NIT; set LQ_{nm} = INI_LQ; reset T_{BSP}.
  - o else /* old neighbor */
    - ⋆ if n's degree changed
      - – do LB-LQE; update δ; reset T_{BSP}
    - ⋆ else (n's degree unchanged)
      - – do LB-LQE
- else if (n is green)
  - o if (new neighbor)
    - – add to NIT; set LQ_{nm} = INI_LQ; reset T_{BSP}.
  - o else /* old neighbor */
    - – do LB-LQE; update δ if changed.
    - ⋆ if (BEP eligible && LQ_{nm} > LQ_THRESHOLD && VAP_{temp} is empty)
      - – set $VAP_{temp} = n$ and reset T_{STBY} to STBY_TIMEOUT.
- else /* n is black */
  - o if (new neighbor)
    - – add to NIT; set LQ_{nm} = INI_LQ; reset T_{BSP} to BSP_TIMEOUT.
    - – set corresponding T_{RB-LQE} to RB_LQE_TIMEOUT.
  - o else /* old neighbor */
    - – update $\delta(n)$ or cluster size based on header flags.
    - – increment the number of GBMs received during T_{RB-LQE}
    - ⋆ if ($\mathcal{N} \equiv m$ && (LQ_{nm}, LQ_{mn}) > LQ_THRESHOLD)

      /* detected a black neighbor with good LQ */
      - – change color to green

3. *T_{GBM} expired:*

- if VAP_{temp} is set to b
  - – generate a GBM with $\mathcal{N} = b$
  - – reset T_{GBM} to GBM_INTERVAL_WHITE
- else (VAP_{temp} is not set)
  - – generate a GBM with $\mathcal{S} = \mathcal{F} = m$, $\mathcal{N} = \mathcal{D}$ = GLOBAL_BROADCAST, $HC = 1$, $TTL = 2$
  - – reset T_{GBM} to GBM_INTERVAL_WHITE

4. *T_{LB-LQE} expired:*

   – delete corresponding neighbor n from NIT /* n is either white or green */

[redacted]

5. T_{RB-LQE} *expired:* /* unlikely to happen but you never know */
   - run RB-LQE and update NIT for n accordingly.

6. T_{STBY} *expired:*
   - delete VAP_{temp}

7. T_{BSP} *expired:*
   - if $\delta(m)$ is highest
     - change to black
   - else
     - reset T_{BSP}

8. T_{BEP} *expired:*
   - make itself BEP eligible
   - if (a green neighbor n with highest $\delta(n)$ exists && LQ_{nm} > THRESHOLD && VAP_{temp} is not set)
     - set $VAP_{temp} = n$ and reset T_{STBY} to STBY_TIMEOUT

[redacted]

GREEN

1. *Beginning of State at node m:*
   - generate a GBM with $\mathcal{N}$ from its VAP_{list} and corresponding LQ update
   - generate a exponentially distributed random number r with mean GBM_INTERVAL_AVG
   - set $T_{GBM} = \min\{r, \text{GBM\_INTERVAL\_MAX}\}$
   - choose a VAP with the highest LQ from VAP_{list} (δ node ID as tie breakers) as its CP
   - initialize SNT (for BCP)

2. *Received GBM from neighbor n:*

   - if (n is white)
     - o if (new neighbor)
       - add to NIT and set LQ_{nm} = INI_LQ.
     - o else /* old neighbor */
       - do LB-LQE and update $\delta(n)$.
     - o if ($\mathcal{N} \equiv m$ && LQ_{nm} > LQ_THRESHOLD) /* VAP Request */
       - update LQ_{mn}
       - set $\mathcal{N} = n$ and hold this GBM /* for bouncing back to n */
       - change color to black
   - else if (n is green)
     - o if (new neighbor)
       - add to NIT; set LQ_{nm} = INI_LQ.
     - o else /* old neighbor */
       - do LB-LQE; update $\delta(n)$ if necessary.
       - update SNT as long as it has at least two non-white neighbors /* BCP */
       - ⋆ if (m must connect its CP and n && LQ_{nm} > LQ_THRESHOLD) /* BCP decides to connect */
         - suspend BCP and flush SNT
         - set $VAP_{temp} = n$ and reset T_{STBY} to STBY_TIMEOUT
     - o if ($\mathcal{N} \equiv m$ && LQ_{nm} > LQ_THRESHOLD && $HC \equiv 1$) /* VAP Request from n */
       - flush SNT and update LQ_{mn}
       - set $\mathcal{N} = n$ and hold this GBM /* for bouncing back to n */
       - change color to black
   - else /* n is black */
     - o if (new neighbor)
       - add to NIT and set LQ_{nm} = INI_LQ
       - set corresponding T_{RB-LQE} to RB_LQE_TIMEOUT.
     - o else /* old neighbor */
       - update $\delta(n)$ or cluster size based on header flags.

- increment the number of GBMs received from n during T_{RB-LQE} by one
- update SNT as long as it has at least two non-white neighbors /* BCP for B-G-B*/

★ if ($\mathcal{S}$ is not one-hop neighbor)

- update routing table and radius as described in Sec 5.5

★ if ($VAP_{temp} \equiv n$ && $\mathcal{F} \equiv m$) /* n was green and was asked to be black by m due to BCP */

- flush SNT; update LQ_{mn}; change color to black

★ else if ($\mathcal{N} \equiv m$ && $LQ_{nm} >$ LQ_THRESHOLD && $HC \equiv 1$)

/* VAP request from black neighbor n due to maintenance */

- flush SNT and update LQ_{mn}
- set $\mathcal{N} = n$ and hold this GBM /* for bouncing back to n */
- change color to black.

- update SNT as long as it has at least two non-white neighbors /* BCP for B-G-P*/

★ if (m must connect its CP and n && $(LQ_{nm}, LQ_{mn}) >$ LQ_THRESHOLD) /* B-G-B */

- change color to black

3. T_{GBM} *expired:*

- generate a exponentially distributed random number r with mean GBM_INTERVAL_AVG

• if VAP_{temp} is set to b

- generate a GBM with $\mathcal{N} = b$, and corresponding LQ update
- set $T_{GBM} = \min\{r, \texttt{GBM\_INTERVAL\_MAX}\}$

• else (VAP_{temp} is not set)

- generate a GBM with $\mathcal{S} = \mathcal{F} = m$ and choose for $\mathcal{N}$ from VAP_{list} with corresponding LQ update
- set $T_{GBM} = \min\{r, \texttt{GBM\_INTERVAL\_MAX}\}$

4. T_{STBY} *expired:*

- delete VAP_{temp} /* LQ with VAP_{temp} was not good enough */
- resume BCP

5. T_{LB-LQE} *expired:*

- delete corresponding neighbor n from NIT /* n is either white or green */

6. T_{RB-LQE} *expired:*

- run RB-LQE and update NIT for n accordingly.

• if ($n \in VAP_{list}$ && $LQ_{nm} <$ LQ_THRESHOLD

- update routes with n as primary gateway as described in Sec 5.5;
- update radius;

∘ if ($CP \equiv n$)

- choose another VAP from VAP_{list} with highest LQ as CP

- if none, choose a green neighbor g with the highest degree and LQ_{gm} > LQ_THRESHOLD
- change to white if no such neighbor.

BLACK

1. *Beginning of State at node m:*
   - if ($\mathcal{N}$ is set to n) /* need to bounce back the GBM it is holding */
     - forward this GBM with $\mathcal{N} = n$, $TTL = 1$, and $HC = 2$
   - else
     - generate a GBM with $\mathcal{N}$ from its VAP_{list} and corresponding LQ update
     - generate a exponentially distributed random number r with mean GBM_INTERVAL_AVG
     - set $T_{GBM} = \min\{r, \text{GBM\_INTERVAL\_MAX}\}$

2. *Received GBM from neighbor n:*
   - drop if duplicate and do not proceed further
   - if ($\mathcal{S}$ is not one-hop neighbor)
     - update routing table and radius as described in Sec 5.5
   - if ($\mathcal{N} \equiv m$ && $LQ_{nm} >$ LQ_THRESHOLD) /* VAP Request */
     - reset T_{VAP} to VAP_TIMEPOUT
   - if (n is white)
     - o if (new neighbor)
       - add to NIT and set $LQ_{nm} =$ INI_LQ.
     - o else /* old neighbor */
       - do LB-LQE; update $\delta(n)$ and LQ_{nm} if necessary.
   - else if (n is green)
     - o if (new neighbor)
       - add to NIT; set $LQ_{nm} =$ INI_LQ.
     - o else /* old neighbor */
       - do LB-LQE; update $\delta(n)$ and LQ_{nm} if necessary.
   - else /* n is black */
     - o if (new neighbor)
       - add to NIT and set $LQ_{nm} =$ INI_LQ
       - set corresponding T_{RB-LQE} to RB_LQE_TIMEOUT.
     - o else /* old neighbor */
       - update $\delta(n)$ or cluster size based on header flags.
       - increment the number of GBMs received during T_{RB-LQE} by one

       /* n was green and was asked to be black by m after T_{VAP} expired (due to maintenance) */
       - ⋆ if ($VAP_{temp} \equiv n$)
         - update LQ_{mn} and change color to green
   - if ($TTL \equiv 1$)
     - drop it

- else
  - $TTL = TTL - 1$ and $HC = HC + 1$
  - ⋆ if ($HC \equiv 2$)
    - $\mathcal{N} = n$ /* bounce back to n */
  - ⋆ else
    - choose $\mathcal{N}$ from NIT in a round-robin manner;
  - forward this GBM

3. T_{GBM} *expired:*

   - generate a exponentially distributed random number r with mean GBM_INTERVAL_AVG
   - if VAP_{temp} is set to b
     - generate a GBM with $\mathcal{N} = b$, and corresponding LQ update
     - set $T_{GBM} = \min\{r, \texttt{GBM\_INTERVAL\_MAX}\}$
   - else (VAP_{temp} is not set)
     - generate a GBM with $\mathcal{S} = \mathcal{F} = m$ and choose for $\mathcal{N}$ from VAP_{list} with corresponding LQ update
     - set $T_{GBM} = \min\{r, \texttt{GBM\_INTERVAL\_MAX}\}$

4. T_{STBY} *expired:*
   - delete VAP_{temp} /* LQ with VAP_{temp} was not good enough */
   - go to "T_{VAP} expired" /* to select another black/green neighbor */

5. T_{LB-LQE} *expired:*
   - delete corresponding neighbor n from NIT /* n is either white or green */

6. T_{RB-LQE} *expired:*

   - run RB-LQE and update NIT for corresponding neighbor n accordingly.
   - if ($n \in VAP_{list}$ && LQ_{nm} < LQ_THRESHOLD)
     - update routes with n as their primary gateway as described in Sec 5.5;
     - update radius;
       - o if (n was the last VAP)
         - choose a black neighbor with highest LQ as CP
         - if none, choose a green neighbor g with the highest degree and LQ_{gm} > LQ_THRESHOLD
     - change to white if no such neighbor.

7. T_{VAP} *expired:*

   - if (one black neighbor b available && (LQ_{mb}, LQ_{bm}) > LQ_THRESHOLD)
     - change to green and generate a GBM with $\mathcal{N} = b$
     - reset T_{GBM}

[redacted]

- else if (no black but at least one green neighbor available)
  - select n which has the highest degree
  - set $VAP_{temp} = n$ and T_{STBY} = STBY_TIMEOUT;
- else /* no green/black neighbors around */
  - change to white;